

Jurnal Sistem Informasi

Volume 4 Nomor 1 Maret 2009

Pelindung :

Rektor Universitas Kristen Maranatha

Penasehat :

Pembantu Rektor Universitas Kristen Maranatha

Pembina :

Dekan Fakultas Teknologi Informasi Universitas Kristen Maranatha

Ketua Tim Redaksi :

Ir. Teddy Marcus Zakaria, MT

Penyunting Ahli :

Ir. Budi Rahardjo, M.Sc, Ph.D

Jazi Eko Istiyanto, Ph.D

Yudho Giri Sucahyo, Ph.D

Penyunting:

Hapnes Toba, M. Sc.

Doro Edi, ST., M.Kom

Elisabet Setiawan, M.Sc.

Radiant Victor Imbar, S.Kom., MT.

Cristian Ade Candra, ST., MT.

Pelaksana Teknis:

Lea Sepvianty Suharso, SE.

Adriani H. Dewi, SE., MM.

Erico Darmawan Handoyo, S.Kom

PENERBIT (PUBLISHER)

Maranatha University Press

ALAMAT PENYUNTING (EDITORIAL ADDRESS)

Sekretariat Jurnal Sistem Informasi UKM

Jurusan Sistem Informasi, Fakultas Teknologi Informasi

Jl. Prof. Drg. Suria Sumantri, MPH, No. 65 Bandung. 40164

Telp (022) 70753665, Fax (022) 2005915

E-mail: jurnal.si@itmaranatha.org

Website: <http://www.itmaranatha.org/jurnal/jurnal.sistem-informasi>

Jurnal Sistem Informasi UKM merupakan jurnal ilmiah sebagai bentuk pengabdian dalam hal pengembangan bidang Sistem Informasi dan bidang terkait lainnya.

Jurnal Sistem Informasi UKM diterbitkan oleh Jurusan Sistem Informasi Universitas Kristen Maranatha. Redaksi mengundang para profesional dari dunia usaha, pendidikan dan peneliti untuk menulis mengenai perkembangan ilmu di bidang yang berkaitan dengan Sistem Informasi.

Jurnal Informatika UKM diterbitkan 2 (dua) kali dalam 1 tahun pada bulan **Maret** dan **September**. Edisi pertama terbit Maret 2006. Harga berlangganan Rp 50.000.- / eksemplar.

DAFTAR ISI

Volume 4 Nomor 1

1	Implementasi <i>Multitier</i> pada Perusahaan	1 - 15
	Indrajani	
2	Integrasi Enterprise (Studi Kasus: Yayasan Pendidikan “X”)	17 - 32
	Tanti Kristanti	
3	Representasi Grafik Multi-Level Berbasis SVG untuk Aplikasi Rich-Content Majalah Mobile	33 - 48
	Adi Nugroho, Theophilus Erman Wellem, Geuis Puspita Dewi	
4	Pemodelan Sistem Informasi pada CV Cihanjuang Inti Teknik dengan Menggunakan <i>Zachman Framework</i>	49 - 60
	Meliana Christianti, Felly Dias Try	
5	Perangkat Lunak JIT (<i>Just In Time</i>) untuk Memprediksi Resiko Proyek Perangkat Lunak	61 - 74
	Yasmi Afrizal, Agus Harjoko	
6	Aplikasi Helpdesk untuk Pencatatan Masalah dan Solusi Perbaikan Peralatan Komputer	75 - 89
	Teddy Marcus Zakaria, Rina Angelina	
7	Sistem Informasi <i>Training & Development</i> di HRD – PT. X	91 - 110
	Radiant Victor Imbar, Evlin Marcelline Fendrianto	

Ucapan Terima Kasih

Redaksi Jurnal Informatika mengucapkan terima kasih yang sebesar – besarnya kepada mitra bestari yang membantu terwujudnya penerbitan Jurnal Informatika Volume 4 Nomor 1 Maret 2009:

1. Kristoko Dwi Hartomo, M.Kom (Universitas Kristen Satya Wacana)

Implementasi *Multitier* pada Perusahaan

Indrajani

Information System, Bina Nusantara University

email : indrajani@yahoo.com

Abstract

This paper tries to explore how far the implementation of information technology in a company in supporting the company's operational process. A lot of companies struggle to be the best and the foremost in their field, and information technology has unquestionably become the ultimate tool to increase the company's performance to bring it to be the best among others. There have been a lot of success stories from companies that become successful after implementing information technology. The implementation of information technology has to be followed by concepts which are suitable with the needs and situation of the company itself. But not all the companies that implement the technology are able to become the best, a lot of other factors need to be brought into concern. Multi Tier is a concept that can be used to support the performance of a company that implements information technology.

Keywords : implementasi, teknologi, multi tier

1. Pendahuluan

Dalam era komputerisasi sekarang ini, penggunaan komputer menjadi satu kebutuhan yang tak dapat dielakkan. Hampir di setiap rumah akan ditemui komputer, baik itu digunakan untuk bekerja maupun untuk hiburan. Komputer bukan lagi menjadi barang langka atau barang mahal mengingat kebutuhan akan komputer sudah seperti halnya telepon yang telah digunakan sehari-hari. Hal ini juga berlaku untuk perusahaan, kebutuhan akan penggunaan komputer sudah menjadi hal biasa, dimana pada era sebelumnya masih dianggap sebagai barang mewah dimana perusahaan yang kecil tidak membutuhkannya, hanya perusahaan besar yang dapat menggunakannya. Dimana dimasa tersebut komputer digunakan lebih dikarenakan untuk membantu pemrosesan data, misal untuk mengolah data penggajian, pembukuan akuntansi. Komputer lebih digunakan untuk memudahkan operasional di dalam perusahaan.

Dalam perkembangannya, penggunaan komputer menjadi hal yang biasa. Hampir semua perusahaan menggunakan komputer sebagai sarana pendukung bagi beroperasinya suatu perusahaan. Banyak perusahaan yang memanfaatkan komputernya hanya untuk mendukung operasionalisasi. Untuk memanfaatkan semaksimal mungkin penggunaan komputer dalam suatu perusahaan, sering dilakukan dengan cara menggabungkan / menghubungkan beberapa komputer. Dengan demikian resource yang ada dapat digunakan bersama-sama. Resource yang ada pada satu komputer dapat dimanfaatkan oleh komputer lainnya. Jaringan komputer ini kemudian disebut sebagai *Local Area Network* (LAN). Penggunaan LAN ini berkembang pesat, karena banyak keuntungan yang dapat diambil dari pemanfaatan LAN.

Penggunaan jaringan yang pada mulanya mengarah kepada *sharing resource*, mulai dikenal konsep adanya *Server* dan *Workstation*. Pemanfaatan jaringan terus berkembang ke aplikasi yang digunakan dalam suatu perusahaan. Konsep yang berkembang pada awalnya adalah konsep *Client Server (Two - Tier)*. Pada konsep ini, dimana *client (workstation)* akan meminta *server* untuk melakukan penarikan data dan hasilnya diberikan kepada *client (workstation)* yang meminta. Penggunaan *Client Server* ini biasanya mengacu kepada pengambilan data pada database.

Masalah

Permasalahan akan timbul dalam konsep ini :

- Digunakan pada skala yang besar.
Server akan berada pada kondisi *overloaded*, jika diakses oleh jumlah *client* (pengguna) dalam keadaan skala besar. Hal ini dapat terjadi karena permintaan akan proses yang dibutuhkan oleh client menjadi bertumpuk di sisi *server*.
- Jika terjadi perubahan proses yang dilakukan di *server*, maka akan mengakibatkan tenaga yang dikeluarkan untuk mengadakan perubahan proses tersebut ke setiap *client* yang ada. Hal ini dapat dilakukan jika pengguna berada dalam jumlah yang sedikit, namun jika pengguna berjumlah besar, tentunya hal ini menjadi masalah yang harus dihadapi.
- Keamanan data
Dengan menggunakan konsep *two-tier* ini, client langsung mengakses *database*. Hal ini dapat mengakibatkan data menjadi berkurang, karena pengguna dapat langsung mengakses *database server*.

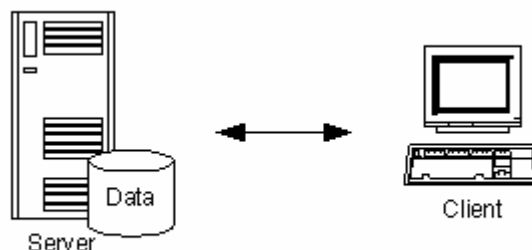
Ruang Lingkup

Client server yang dibahas pada tipe *database server*

2. Pembahasan

Client Server

Pada dasarnya *Client Server* adalah suatu metoda untuk disain dan implementasi dari suatu aplikasi dengan cara membagi ke fungsi-fungsi yang berdasarkan proses yang dilakukan.



Gambar 2.1 *Client Server*

Konsep

Proses Client Server

- *Client Proses*

Client akan mengirimkan pesan untuk proses yang akan dilakukan di *server*. Proses pengiriman pesan ini dapat berupa program (aplikasi). Aplikasi ini dapat berupa isian (*entry*) yang dilakukan oleh *user* (pengguna). Setelah mengirimkan pesan proses, maka *client* akan menunggu hasil/status dari proses yang dilakukan oleh *server*. Setelah mendapatkan hasil proses/status, maka *client* akan masuk ke proses selanjutnya. Dan menampilkan hasil dari proses dari *server* untuk ditampilkan kepada pengguna tersebut.

- *Server Proses*

Saat *server* menerima pesan untuk melakukan suatu hal, maka *server* akan memproses permintaan tersebut dan mengirimkan hasil/status proses tersebut.

Dampak dari penggunaan konsep *Client Server*, antara lain akan meningkatkan *performance* dari *client* (karena *client* hanya mengambil hasil / status dari *server* dan memprosesnya lebih lanjut). Tetapi ada hal lain yang harus diperhatikan, yaitu masalah *network traffic*. Jika *network traffic* dalam keadaan yang tinggi, maka konsep *client server* akan memperlambat proses yang ada.

Dalam zaman teknologi informasi yang berkembang saat ini serta diikuti oleh tingkat kompetisi yang tinggi, maka pemanfaatan teknologi yang tepat guna dan efisien merupakan hal mutlak yang harus dilakukan oleh perusahaan jika ingin maju dan memimpin pasar di bidangnya. Teknologi dapat dimiliki oleh semua perusahaan tapi konsep dan arsitekturnya berbeda-beda. Perusahaan yang dapat menerapkan arsitekturnya dengan tepat, maka perusahaan tersebut akan lebih maju. Kadang kala perusahaan yang salah menerapkan teknologi atau arsitektur akan mengakibatkan ketidakefisien dalam perusahaan tersebut.

Dengan teknologi networking yang berkembang saat ini, internet cukup menjanjikan banyak keuntungan yang dapat diperoleh. Cukup hanya dengan browser, tanpa instalasi aplikasi, maka user dapat menggunakan sistem yang diinginkan.

Tetapi yang menjadi masalah bagaimana jika user yang mengakses aplikasi yang sama dalam jumlah besar, apalagi jika aplikasi tersebut dalam bentuk transaksi. Tentunya hal ini menjadi satu permasalahan yang harus dicari jalan keluarnya. Hal seperti ini menjadi lebih penting lagi apabila customer yang menggunakan aplikasi tersebut. Apabila customer kecewa dengan aplikasi yang ada, akan ada kemungkinan customer tersebut akan berpindah ke perusahaan lain yang sejenis, lain halnya jika perusahaan tersebut merupakan perusahaan monopoli. Teknologi Informasi bukan hanya sebagai pendukung proses perusahaan tetapi sudah menjadi salah satu hal strategis buat perusahaan, dengan demikian hal tersebut harus menjadi perhatian.

Pada awal tahun 1990, banyak perusahaan melakukan reengineering infrastruktur bisnis mereka untuk melakukan perubahan konsep teknologi informasi ke konsep client-server application. Hal ini bertujuan untuk meningkatkan keunggulan dan produktifitas dari perusahaan, khususnya hal-hal yang berhubungan dengan transaksi dan juga langsung digunakan oleh customer dari perusahaan tersebut. Konsep ini memungkinkan bisnis berjalan dengan lebih

efisien dalam berkolaborasi, pemanfaatan informasi bersama-sama. Dengan demikian setiap user tidak harus menyimpan informasi di masing-masing workstation tetapi dapat ditempatkan pada server, sehingga user yang lain dapat melihat informasi tersebut pada server. Sebagai contoh para eksekutif ingin melihat Executive Information System (EIS), cukup melalui workstation mereka. Jika terjadi perubahan data, maka saat itu juga dapat dilihat hasilnya.

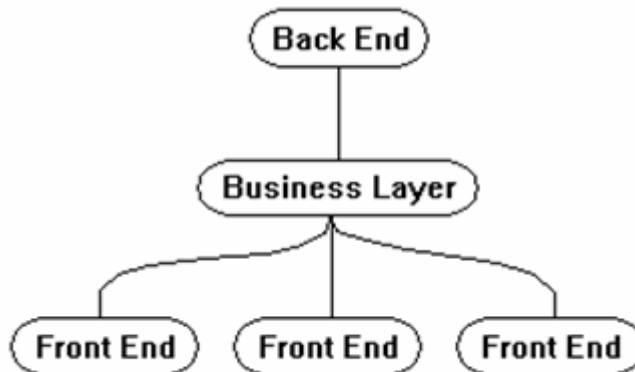
Dalam perkembangannya, seiring dengan berkembangnya organisasi, konsep client server (two - tier) mengalami keterbatasan.

Dengan jumlah user yang sangat besar, lingkungan dengan banyak database, dan suatu jaringan network yang tidak aman, maka aplikasi 2-tier dapat mengalami beberapa keterbatasan, antara lain :

- Database harus selalu mempertahankan koneksi pada tiap client yang aktif. Koneksi-koneksi ini akan banyak mengkonsumsi resource server dan jaringan network yang ada, sehingga pada akhirnya akan mengurangi kemampuan server, seiring dengan bertambahnya jumlah user.
- Gangguan dapat terjadi ketika dalam database ada banyak client yang mengakses data yang sama pada saat bersamaan. Untuk mencegah terjadinya korupsi data, setelah seorang client meminta ijin untuk mengakses suatu potong data tertentu (misal suatu baris tertentu), maka database akan “mengunci” (lock) data tersebut untuk mencegah client lain untuk mengaksesnya pada saat yang bersamaan. Client yang lain harus menunggu sampai database melepaskan kunciannya sebelum mereka dapat melanjutkan dengan pekerjaan mereka.
- Model sekuriti yang digunakan pada system 2-tier tidak dapat bekerja dengan baik di luar jaringan LAN (Local Area Network) yang aman dan terpercaya. Keamanan pada 2-tier berfokus pada pemberian ijin pada user, apakah mereka berhak atau tidak untuk mengakses data. Begitu administrator memberi ijin pada seorang user untuk mengakses data suatu tabel, maka user tersebut pada dasarnya dapat melakukan apa saja terhadap data tersebut termasuk melakukan perubahan dan penghapusan data.
- Sangatlah sulit untuk menggunakan kembali suatu logika aplikasi pada arsitektur 2-tier secara luas, karena aplikasi terikat dengan sangat erat kepada sistem database dan format tabel tertentu. Menggunakan kembali logika aplikasi 2-tier biasanya berarti memotong-dan-menempel (cut-and-paste) kode antar aplikasi.
- Sistem 2-tier hanya dapat mengakses satu database pada satu waktu. Akses kepada sistem database yang lain, aplikasi mainframe, atau resource lain harus dilakukan melalui suatu gerbang (gateway), yang pada gilirannya akan menciptakan beberapa masalah baru serta memakan resource yang tidak sedikit pada jaringan.
 - Yang harus juga diperhatikan adalah, bahwa karena adanya lock-contention, kinerja aplikasi 2-tier dapat memburuk dengan cepat, khususnya pada saat diakses oleh banyak user secara bersamaan. Karena resolusi “penguncian” (lock) tidak tergantung pada kemampuan server, maka meski menambah server database baru yang lebih cepat pun tidak akan banyak membantu meningkatkan kinerja sistem.

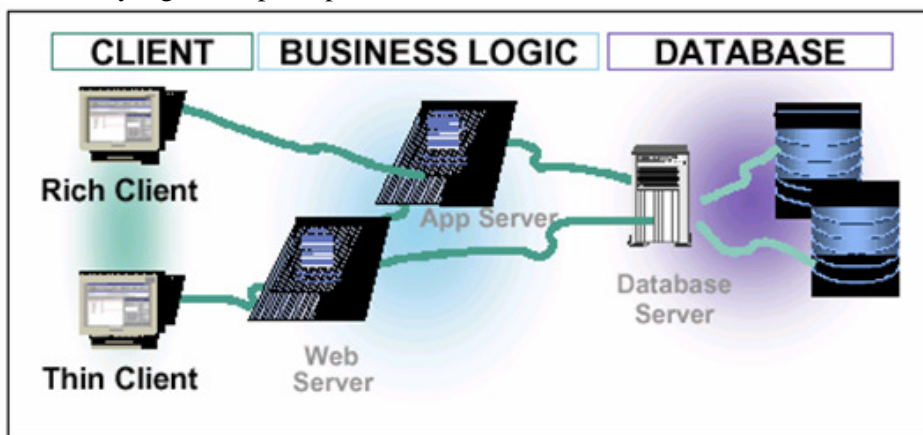
Untuk menanggulangi keterbatasan ini, pada aplikasi-aplikasi besar menggunakan konsep three-tier yang terdiri dari business logic sebagai middleware antara user-interface dan data-repository. Dengan konsep ini, aplikasi akan lebih mudah untuk dikelola karena pembagiannya lebih menolong untuk meningkatkan performance dari sistem.

Dalam konsep three-tier ini akan didapatkan kemudahan update aplikasi, kesanggupan untuk melayani informasi atau transaksi dalam jumlah yang besar, dikenalkan adanya konsep 3-Tier. Konsep ini mengenalkan adanya tiga layer yang akan melakukan sesuai dengan sesuai dengan fungsinya.



Gambar 2.2 Konsep sederhana *three tier*

Dalam perkembangan selanjutnya, dengan menggunakan konsep pendekatan CRM (Customer Relationship Management), maka istilah front / Back End dihilangkan, dan dapat dikatakan semua merupakan front End. Untuk itu arsitektur yang ada dapat seperti berikut ini :



Gambar 2.3 Contoh pengaplikasian *three-tier*

Client akan melakukan proses penginputan yang hasilnya akan diproses oleh business layer (Middle Tier) untuk dilakukan validasi dan setelah diproses pada middle layer ini akan dilanjutkan untuk melakukan koneksi ke database (jika perlu) dan hasilnya akan dikembalikan kepada client. Dengan demikian proses di client akan menjadi lebih ringan karena proses validasi akan dilakukan pada

business layer termasuk koneksi ke database server. Dengan demikian resource yang ada pada client dapat digunakan untuk hal lain yang mungkin dapat meningkatkan performance sistem. Hanya hal-hal yang berhubungan dengan data proses akan dilakukan pada database server sehingga kinerja kerja dari database server ini diharapkan menjadi lebih baik.

Microsoft mengeluarkan konsep arsitektur Three-Tier dengan pendekatan service, yaitu :

- User Services
- Business Services
- Data Services

User Services

Dalam layer services ini, user dimungkinkan untuk memanipulasi data dan melakukan input data. Interface yang digunakan dapat berupa aplikasi biasa atau dengan web-base application (browser). Secara garis besar fungsi dari layer ini adalah :

- Mengumpulkan informasi dari user
- Mengirim informasi tadi ke Business Services untuk diproses
- Menerima hasil dari Business Services
- Menampilkan hasil proses ke user

Business Services

Layer Dalam services ini, terdapat aturan bisnis atau aturan data. Semua aturan yang ada akan ditempatkan di sini, dan akan digunakan oleh semua client yang terhubung ke services ini. Dalam layer ini dapat ditempatkan aturan perusahaan, aturan pemerintah yang diperlukan oleh aplikasi yang memanggilnya. Secara garis besar fungsi dari layer ini adalah :

- Menerima masukan dari user services
- Berinteraksi dengan data user services untuk melakukan operasi bisnis yang ditugaskan padanya secara otomatis (misal : menghitung jumlah pajak)
- Mengirim hasil yang sudah diproses ke user services

Data Services

Layer ini berhubungan langsung dengan database dan di simpan dalam storage tetap (media penyimpanan). Data dapat diakses melalui business services dan data services itu sendiri. Secara garis besar fungsi dari layer ini adalah sebagai berikut :

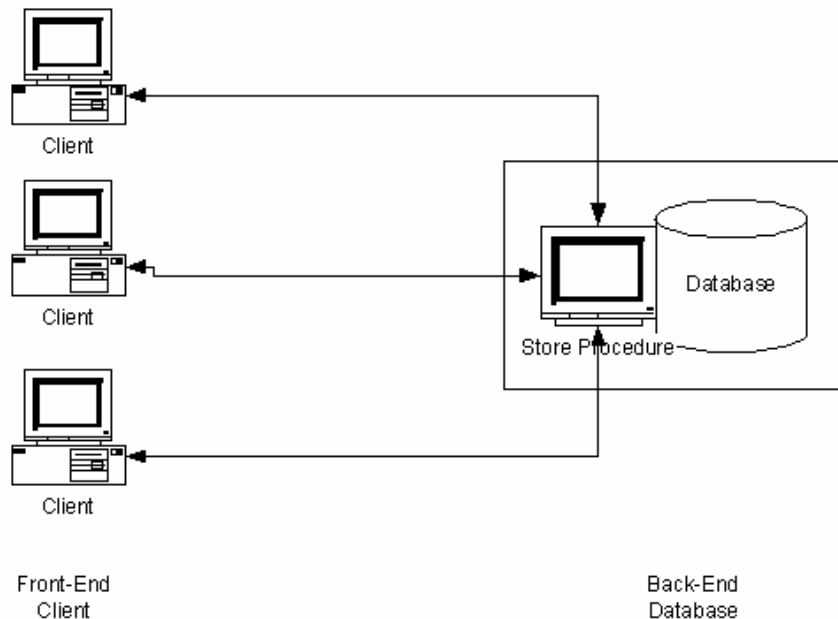
- Tempat pengambilan data
- Tempat mengatur data (termasuk menjaga integritas data)

Data services ini banyak bentuk dan ukurannya, termasuk RDBMS (*Relational Database Management System*) seperti SQL Server, E-mail Server (Microsoft Exchange Server)

Pembagian tiga services ini tidak berarti harus terdapat 3 buah hardware yang berbeda, bisa dilakukan dengan penggabungan dua services ke dalam satu hardware (komputer). Dalam konsep ini lebih ditekankan kepada fungsi dari setiap bagian yang ada.

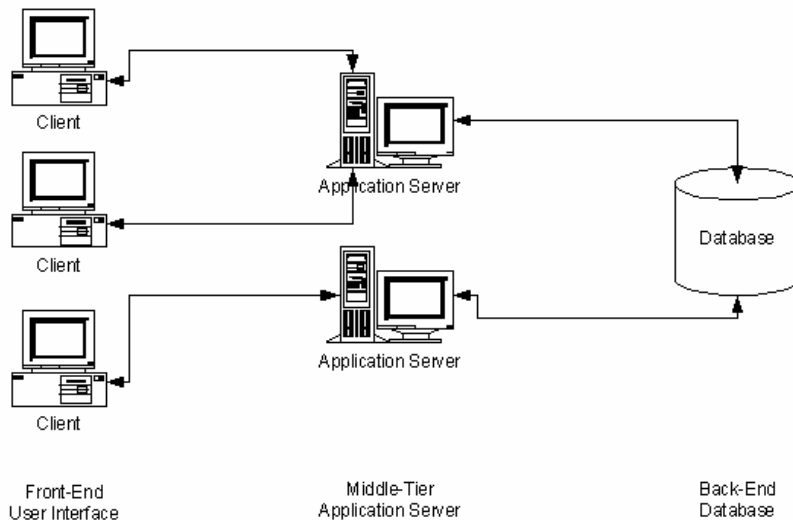
Keuntungan yang dapat diperoleh dengan penerapan konsep Three-Tier :

- Meningkatkan performance dari aplikasi
Dengan adanya pembagian kerja sesuai dengan fungsi yang telah ditetapkan, maka proses yang dapat ditanganipun akan meningkat. Dengan demikian penyediaan Graphical User Interface (GUI) yang user friendly dapat lebih memungkinkan. Karena sebelumnya konsentrasi lebih ditujukan kepada beban client yang cukup berat, dengan didistribusikannya beban tersebut, maka hal lain dapat dimasukkan ke client untuk menunjang kinerja kerja dari aplikasi tersebut.
- Scalability
Arsitektur ini dapat dengan cepat dan mudah menaikkan jumlah transaksi user tanpa perlu perubahan besar pada investasi hardware dan software. Misalkan pada suatu client server yang 2-tier yang meletakkan prosedur penyimpanan order pada database server. Ketika volume transaksi membesar, database server menjadi pelan. Untuk itu menaikkan unjuk kerja kembali, maka pilihan untuk penambahan database server sulit untuk dilakukan.



Gambar 2.4 Arsitektur *Two Tier*

Pada sistem 3-tier, masalah ini dengan mudah dapat dipecahkan, yaitu dengan cara menambahkan middle-tier server. Setiap server menjalankan program business server yang sama. Tidak menjadi masalah client mana yang dilayani, karena setiap client dapat melakukan dengan koneksi server yang manapun, ketika yang satunya sibuk.



Gambar 2.5 Arsitektur *Three Tier*

- *Reuseability*

Business rules yang telah didefinisikan pada Middle-Tier dapat digunakan oleh aplikasi lain yang mempunyai karakteristik yang sama, bahkan jika memungkinkan dapat dimodifikasi sehingga beberapa aplikasi dapat menggunakan business rules yang sama. Hal ini menyebabkan kemudahan dalam maintenance. Jika ada perubahan rules dalam suatu aplikasi tertentu, maka developer cukup hanya merubah business rules yang ada pada middle-Tier, maka aplikasi sudah dapat berfungsi untuk rules yang baru, sedangkan jika menggunakan konsep 2-Tier, maka di setiap client yang ada mesti dirubah satu per satu. Hal ini akan memakan waktu dan biaya yang tidak sedikit.

- *Security*

Untuk perusahaan besar dengan jumlah karyawan yang besar, keamanan data menjadi hal yang sangat penting yang tidak boleh dilupakan. Dengan adanya perubahan data yang tidak diinginkan dapat menyebabkan kerusakan sistem maupun kerugian yang tidak sedikit. Masalah security juga menjadi hal yang dapat ditingkatkan dengan menggunakan konsep Three-Tier ini. Pengaksesan data hanya dapat dilakukan melalui middle-Tier, user tidak dapat langsung masuk ke database seperti halnya pada konsep Two-Tier. Dalam konsep two-tier client akan langsung berhubungan database server, dengan ditambahkannya satu layer akan meningkatkan tingkat keamanan data.

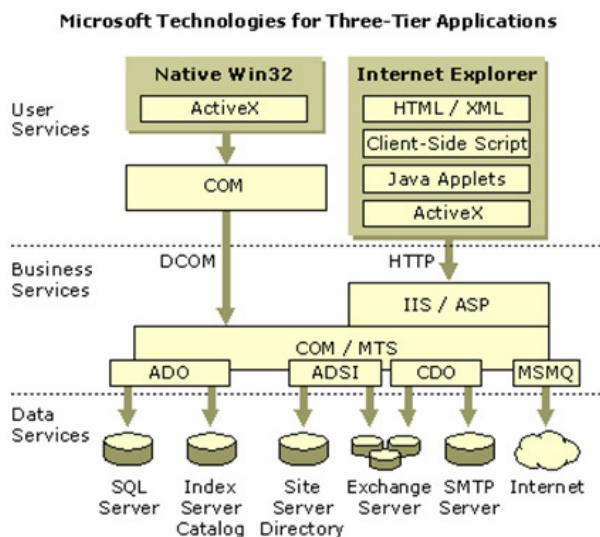
Suatu indikasi dari fleksibilitas dari arsitektur 3-tier ini adalah kenyataan bahwa client dan database server tidak harus merupakan suatu perangkat komputer yang biasa dibayangkan. Client tidaklah harus merupakan suatu workstation. User dapat menjalankan aplikasi dari Macintosh, atau dari tombol tekan pesawat telepon, ataupun dari Automated Teller Machine (ATM). Kesemuanya dikoneksikan kepada business process server yang sama, dan menjalankan suatu kumpulan aturan bisnis yang sama. Begitu juga dengan database server, dapat berupa sesuatu yang bukan produk SQL. Sebab business process layer menyajikan pada client dengan suatu

pandangan logis terhadap data, menghalangi akses langsung client dari dan ke database server. Back end server dapat berupa IMS, VSM atau suatu sumber data yang real time. Bentuk penyimpanan data pada back end server ini tidak memberikan pengaruh pada proses bisnis. Pada terapannya, trend yang ada pada saat ini adalah menggunakan 3-tier sebagai suatu front-end untuk sistem non-relational, misal legacy mainframe system.

Untuk pengembangan lebih lanjut, kini dikembangkan arsitektur many-tiered. Aplikasi didistribusikan ke lebih dari tiga platform, yang biasanya dilakukan dengan membagi proses bisnis tersebut. Arsitektur ini dapat juga disebut 4-tier. Ada juga variasi yang bertujuan sebagai suatu penyederhanaan. Pembagian ketiga tier ini dapat juga tidak dilakukan secara fisik diletakkan di tiga sistem komputer terpisah yang saling dihubungkan. Akan tetapi diletakkan pada satu atau dua komputer saja. Yang penting adalah aplikasi-aplikasi tersebut tersegmentasi secara logis dan saling tidak bergantung, tetapi dapat saling berkomunikasi dan bertukar message dan data.

Untuk client, banyak jenis platform yang dapat digunakan, demikian juga halnya dengan database server. Untuk Midde-Tier microsoft mengeluarkan productnya yang disebut sebagai Microsoft Transaction Server (MTS). Produk ini yang akan menjadi middle-tier bagi aplikasi yang menggunakan konsep three-tier.

Microsoft Transaction Server merepresentasikan sebuah teknologi yang handal untuk mengembangkan dan membuat aplikasi dengan konsep 3-tier berdasarkan pada teknologi Component Object Model (COM). Diantara banyak peran, web telah menjadi alat yang ideal untuk menjadi aplikasi kepada user. Browser menjadi jalan yang paling mudah untuk merepresentasikan user-interface yang dinamis, apalagi biaya perawatannya yang tidak terlalu besar. Internet atau intranet sekarang juga sudah menjadi bagian dari seluruh dunia dan memungkinkan untuk mengubah desktop menjadi lebih virtual. Presentasi dan masukan dari user adalah hanya sebagian dari keseluruhan aplikasi. Dan sekarang mekanisme web server sudah bisa untuk menerima input dari user, memanggil sebuah aplikasi atau sebuah script, dan mengembalikan sebuah response.



Gambar 2.6 Arsitektur MTS

Microsoft Transaction Server (MTS) yang termasuk dalam *Microsoft Internet Information Server 4.0* (IIS) adalah ideal untuk membangun sebuah aplikasi berkonsepkan 3-tier dengan front-end nya yang berbasis web. IIS menyediakan Active Server Pages (ASP) sebagai mekanisme aplikasi agar web yang dihasilkan dapat berinteraksi dan dinamis. ASP adalah sebuah file teks biasa yang berisi kombinasi dari sintaks-sintaks HTML standard dan script-script yang berbasis Visual Basic dan JavaScript. Di dalam ASP dapat dilakukan penghitungan-penghitungan dasar, pengaksesan database melalui interface ODBC dan yang terutama ASP dapat memanggil komponen-komponen yang berjalan di dalam MTS. Ketika ASP memanggil komponen yang ada di MTS, komponen-komponennya dapat melakukan :

- Melakukan penghitungan pada bagian logika dari suatu aplikasi.
- Mengakses satu atau lebih database melalui ODBC 3.0 seperti *SQL Server 7.0* atau *Oracle 8i*.
- Memanggil komponen-komponen lain, melakukan bagian-bagian dari logika aplikasi dan memperbolehkan pemakaian ulang dari komponen. Pada kenyataannya, perusahaan-perusahaan dapat mengurangi waktu pengembangan dengan meng-compose aplikasi-aplikasi dari pool of prebuilt dan purchased components.

Karena MTS menyediakan fasilitas transaksi secara otomatis, jika terjadi error dari komponen yang ada di MTS ataupun error pada script dari ASP, MTS akan melakukan roll back dari semua perubahan yang dibuat ke database. MTS menyediakan proteksi terhadap keintegritasian data.

Dengan MTS, pengembang dapat menggunakan kemampuannya yang telah dipunyai dengan tools seperti Visual Basic untuk membuat aplikasi 3-tier. Dengan mengikuti sedikit contoh dari aturan-aturan, para pengembang dapat membuat komponen-komponen dalam bentuk file dynamic-link libraries (DLLs), yang dijalankan pada server middle-tier dibawah kontrol dari MTS. Dari semua itu maka terdapat beberapa keuntungan, seperti :

- Kemampuan untuk mengakses database-database dan resource yang lain dengan proteksi penuh pada transaksi yang dilakukan.
- Menyederhanakan pengembangan komponen melalui MTS Explorer dengan interface drag-and-drop saja.
- Pengaturan komponen yang mudah melalui MTS Explorer.
- Mempunyai kemampuan untuk memanggil objek-objek yang sama dari IIS dan ASP.

Dengan penerapan konsep three-tier ini, banyak manfaat yang dapat diambil dalam mempermudah atau meningkatkan performance dari aplikasi yang ada. Dalam setiap konsep yang ada, bukan berarti konsep tersebut tidak mempunyai kelemahan. Demikian pula halnya dengan konsep three-tier ini, tidak semua aplikasi dengan menerapkan konsep ini kinerjanya menjadi lebih baik. Ada beberapa hal yang harus diperhatikan agar penerapan konsep ini tidak menjadi penyebab gagalnya / memperlambat beroperasi suatu aplikasi.

Besarnya / rumitnya business rules yang digunakan.

Jika dalam aplikasi yang digunakan hanya merupakan aplikasi yang tidak terlalu besar dan tidak terdapat transaksi. Jika aplikasi tersebut menggunakan konsep three-tier akan mengakibatkan aplikasi menjadi lebih lambat. Hal ini disebabkan karena untuk dapat berkomunikasi baik itu antara client dengan middle-tier dan antara middle-tier dengan database server. Dalam komunikasi ini akan melalui jaringan network yang ada dan proses ini membutuhkan waktu. Dengan kata lain waktu yang dapat diperoleh antara selisih lebih cepatnya proses pada di middle-tier tidak sebanding dengan lambatnya proses pada jaringan network yang ada.

Resource Jaringan

Jika sudah bermain dalam jaringan network, maka harus memperhatikan resource yang digunakan dalam konsep ini. Hal ini sangat berpengaruh dalam penerapan (implementasi) konsep ini. Hal lain yang dapat menjadi pertimbangan adalah apakah biaya yang dikeluarkan untuk konsep three-tier ini akan setimpal dengan hasil yang akan diperoleh, baik itu dari sisi security, performance dan kevaliditas data yang dihasilkan serta kemudahan maintenance sistem. Jika ternyata setelah dianalisa diketahui penerapan three-tier tidak efektif dan tidak membawa hasil bagi sistem aplikasi, maka tidak perlu dipaksakan untuk itu.

Implementasi

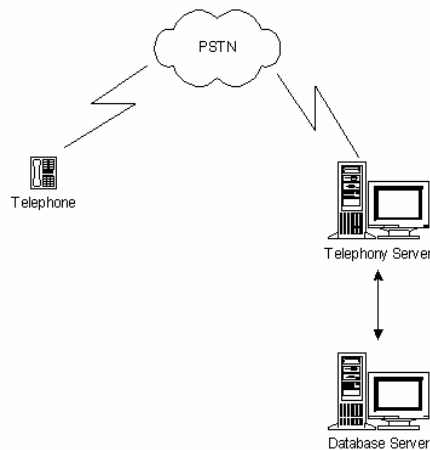
Pada suatu universitas dalam setiap semesternya melakukan proses registrasi. Dalam proses registrasi tersebut, terdapat beberapa tahap proses yang perlu dilakukan. Antara lain persiapan cara mahasiswa untuk melakukan registrasi. Misalnya universitas tersebut menyediakan 2 cara online yang dapat dilakukan oleh mahasiswa, melalui Telepon yang dikenal dengan Phone Service (PS) dan melalui Internet (web) yang dikenal dengan KRS Online.

Dalam proses registrasi ini mahasiswa dapat menyusun jadwal kuliahnya sendiri beserta hari, jam dan mata kuliah yang ingin diikuti. Setiap kelas mempunyai keterbatasan kapasitas yang dapat ditampung. Dalam suatu mata kuliah juga mempunyai prasyarat yang harus dipenuhi terlebih dahulu sebelum mata kuliah tersebut dapat diambil. Setiap mahasiswa dibatasi jumlah maksimal mata kuliah yang dapat diambil berdasarkan jumlah SKS.

Pengisian dapat dilakukan baik melalui PS atau Web, namun tidak boleh dilakukan melalui kedua cara tersebut pada waktu bersamaan, dan kedua cara tersebut harus dapat saling berkomunikasi melalui database. Konsep yang digunakan menggunakan two-Tier.

Berikut ini arsitektur dari masing-masing media yang ada :

Arsitektur Phone Service

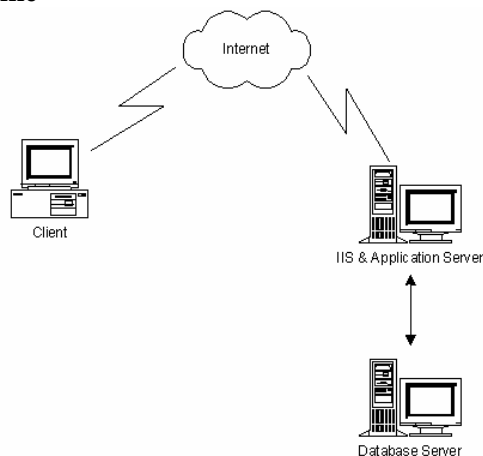


Gambar 2.7 Arsitektur PS

Mahasiswa dapat menelpon nomor telephone yang telah ditentukan sebelumnya oleh universitas, maka mahasiswa tersebut akan disambungkan ke telephone server yang telah mempunyai aplikasi telephony. Input yang dilakukan oleh mahasiswa akan diterima oleh telephony server dan akan diproses oleh telephony server. Jika Telephony server membutuhkan data, maka akan dilakukan koneksi ke database server dan data akan diambil dan diproses lebih lanjut di telephony server. Hal ini akan cukup membebani Telephony server, karena hampir semua proses dilakukan di server tersebut, dari proses input sampai dengan business rules yang ada. Validasi juga mencakup kapasitas kelas, jika kapasitas sudah habis maka langsung akan ada pesan penolakan terhadap kelas tersebut. Semua ini dilakukan secara online.

Untuk mengurangi beban tersebut, sebagian validasi database dilakukan pada database server, namun validasi tersebut hanya sebagian kecil dari proses registrasi.

Arsitektur KRS Online



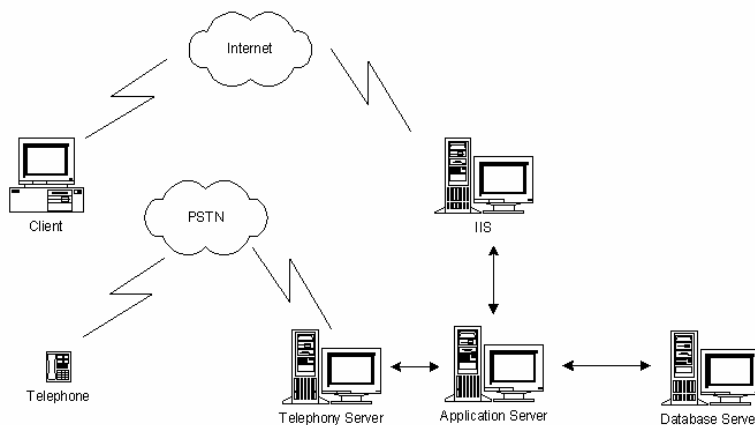
Gambar 2.8 Arsitektur KRS Online

Mahasiswa dapat melakukan proses registrasi dengan cara masuk ke web site universitas dan mengisi mata kuliah yang ingin diambil. Jika ada penolakan mata kuliah, maka akan ada pesan kesalahan kemudian mahasiswa dapat memperbaiki dan mengirimkannya kembali ke server.

Setelah server menerima paket data yang dikirimkan oleh mahasiswa melalui aplikasi web, maka paket data tersebut akan diproses pada application server. Jika proses tersebut berhasil, maka akan dilakukan penyimpanan data ke database, jika ternyata ada penolakan, maka akan ada pesan kesalahan kepada user melalui internet dan server akan kembali menerima paket data dan memprosesnya kembali.

Jika mempunyai 2 cara untuk melakukan proses registrasi dan masing-masing dari kedua cara tersebut melakukan proses validasinya sendiri-sendiri, ada kemungkinan dimana melalui PS diterima tetapi melalui web bisa ditolak, atau juga kondisi sebaliknya. Hal ini dikarenakan pada PS proses validasi dilakukan pada Telephony Server tetapi pada Web validasi dilakukan di Application server. Hal seperti ini mempersulit pengupdatean validasi. Jika ada perubahan, maka harus dilakukan pada dua tempat yang berbeda.

Setelah melakukan analisa terhadap permasalahan yang ada, maka diperlukan adanya perubahan arsitektur pada PS dan KRS Online untuk mendukung proses registrasi menjadi lebih efisien. Setelah dilakukan tahapan disain, maka berikut ini adalah arsitektur PS dan KRS Online yang baru dengan menggunakan konsep three-tier :



Gambar 2.9 Arsitektur PS dan KRS Online dengan *three tier*

Dengan arsitektur yang baru ini, semua validasi baik itu untuk PS maupun untuk KRS Online dilakukan pada server yang sama, yaitu pada application server (middle-tier). Telephony server dan IIS sebagai client bagi Application Server yang nantinya akan melakukan koneksi ke database server.

Pada saat mahasiswa melakukan proses registrasi melalui PS, maka Telephony Server akan menerima semua input dan melakukan proses validasi. Proses ini dilakukan pada Application Server (yang berfungsi sebagai middle-tier), setelah diproses maka akan ada hasil dari proses tersebut. Hasil ini akan dikirimkan kembali ke Telephony Server dan hasil ini yang akan diberikan kepada mahasiswa.

Pada saat mahasiswa melakukan input berupa mata kuliah yang ingin diambil, maka telephony server akan memanggil validasi yang ada pada application server beserta nim dan mata kuliah yang diambil. Selanjutnya Application Server akan melakukan pengecekan terhadap mata kuliah tersebut. Pengecekan ini meliputi banyak hal, antara lain hak sks yang masih diperbolehkan untuk diambil, kapasitas kelas, jadwal kuliah dan lain sebagainya. Jika terdapat kesalahan dalam penginputan tersebut, maka application server akan memberikan kode error yang akan dibaca oleh telephony server dan akhirnya akan diberikan kepada mahasiswa.

Proses yang hampir sama kan dialami oleh mahasiswa yang melakukan proses registrasi melalui internet (KRS Online). Paket data yang dikirimkan tersebut akan diproses melalui application server dan melakukan validasi yang sama dengan PS dan jika ada kesalahan maka akan dikeluarkan kode error yang akan diproses oleh browser pada client (mahasiswa), mahasiswa dapat kembali memperbaiki kesalahan tersebut dan mengirimkannya kembali dan diproses ulang seperti sebelumnya.

Jika pada proses selanjutnya ada perubahan validasi untuk proses registrasi, maka cukup hanya merubah validasi yang ada di application server, dengan demikian pada PS dan KRS Online akan mengikuti rules yang telah direvisi tersebut.

3. Penutup

Simpulan dari implementasi multi tier ini adalah sebagai berikut :

- Penerapan konsep three-tier sangat membantu untuk aplikasi yang banyak menggunakan transaksi serta dalam jumlah user yang banyak pada saat bersamaan.
- Penggunaan konsep three-tier ini tidak dapat langsung diterapkan pada sistem sistem aplikasi yang ada, tetapi melihat kebutuhan akan sistem itu sendiri.
- Peningkatan performance, security serta kemudahan dalam maintenance menjadi hal pokok dalam konsep three-tier.
- Investasi yang diperlukan untuk konsep three-tier menjadi lebih besar jika dibandingkan dengan two-tier, tetapi manfaat yang diperoleh lebih besar jika dibandingkan dengan biaya yang harus dikeluarkan.

Daftar Pustaka

- [1]. Process of Change : **Tier Three** Design. Availabe : 2009.
<http://blogs.msdn.com/bobreb/archive/2007/08/20/tier-three-design.aspx>.
- [2]. <<http://blogs.msdn.com/bobreb/archive/2007/08/20/tier-three-design.aspx>>. Accessed : 19 March 2009
- [3]. **Three Tier** Architecture. Availabe : 2009
- [4]. <http://social.msdn.microsoft.com/Forums/en-US/netfxnetcom/thread/b3a54e52-3c34-4fd2-b299-1cc0e503a7f7/>
- [5]. <http://social.msdn.microsoft.com/Forums/en-US/netfxnetcom/thread/b3a54e52-3c34-4fd2-b299-1cc0e503a7f7/>. Accessed : 19 March 2009
- [6]. [msdn2009] **Three-tier** Application Model . Availabe : 2009.
<http://msdn.microsoft.com/en-us/library/aa480455.aspx>. <

- <http://msdn.microsoft.com/en-us/library/aa480455.aspx>>. Accessed : 19 March 2009
- [7]. **Three Tier** : How do **three** layers communicate ? Availabe : 2009
- [8]. <http://social.msdn.microsoft.com/Forums/en-US/architecturegeneral/thread/4f417a46-ceed-4c15-a948-27418c0736b3/>
- [9]. <<http://social.msdn.microsoft.com/Forums/en-US/architecturegeneral/thread/4f417a46-ceed-4c15-a948-27418c0736b3/>> . Accessed : 19 March 2009
- [10]. Using a **Three-Tier** Architecture Model . Availabe : 2009
<http://msdn.microsoft.com/en-us/library/ms685068.aspx> <
<http://msdn.microsoft.com/en-us/library/ms685068.aspx> >
Accessed : 19 March 2009

Integrasi Enterprise (Studi Kasus: Yayasan Pendidikan “X”)

Tanti Kristanti

Jurusan Teknik Informatika, Fakultas Teknologi Informasi
Universitas Kristen Maranatha
Jl. Surya Sumantri No. 65, Bandung 40164
e-mail: tantikristanti02@yahoo.com

Abstract

Ownership of information system at one particular company is expected to support their totally business process, which is not limited to just certain business function but can give benefit to a number of functional areas. This tendency claims to the requirements to integrate the function oriented system in order to make them in line with business process. Yayasan Pendidikan “X” as educational institution has also a number of business function supported by information system. But in its growth, management of the system is conducted independently by a number of organizational unit regardless of business requirement so that emerging “islands” of information system as impact from the system that having stovepipe character and unable to communicate one with another. Enterprise integration is the answer for the problems resulted from the system that has differ platform which cannot give optimal benefit to company.

To be able to obtain holistic benefit for business, hence integration project will preceded by analysis activities to capture enterprise present condition. Analysis is aimed to understand “X” current condition (as-is), started with business modeling, defines what is in place today for application system and supporting technology that result information resource catalog. Business modeling by using Porter Chain Value as well as Four Stage Life Cycle Business System Planning is expected to understand functional areas owned by “X” so that yielded architecture that is not impressionable by internal and external changing environment. Information resource catalog analysis constitute integration project that can be manage in an optimal fashion without depending on certain technology. Pursuant to result of analysis from current condition, hence major kinds of data, technology and application for future needs can be determined (to-be). As-is analysis and to-be definition will result some gaps that constituting policies for integration project.

Integration is also triggered by business drivers and requirements. The business drivers and requirements which are triggering the needing of integration project in “X” are the needs to increase business efficiency and competitiveness and to improve customer satisfaction for both internal and external customer of “X”. Integration architecture development covers technical, service, information and business process integration. The results from this architecture then are used as basis for integration implementation strategy.

Keywords: enterprise integration, enterprise architecture planning.

1. Pendahuluan

Informasi telah menjadi salah satu aset yang penting dalam perusahaan, oleh karenanya perusahaan melakukan berbagai upaya untuk dapat mengumpulkan serta mengolah data menjadi informasi yang bermanfaat bagi kegiatan bisnisnya. Salah satu upaya tersebut adalah dengan membangun sistem informasi yang didukung

oleh teknologi informasi. Namun, perusahaan pada umumnya membangun sistem informasi/teknologi informasi (SI/TI) hanya untuk mendukung fungsi bisnis tertentu, sedangkan suatu perusahaan terdiri atas banyak fungsi bisnis, yang masing-masing fungsi tersebut juga ternyata memerlukan dukungan SI/TI. Tidak jarang, sistem informasi yang dibangun tidak selaras dengan tujuan bisnis secara menyeluruh karena sistem informasi dalam perusahaan seringkali hanya dipandang sekedar pengadaan teknologi yang mampu memenuhi kebutuhan jangka pendek dan untuk mengikuti *trend* teknologi. Pengembangan sistem yang hanya memperhatikan kebutuhan fungsi tertentu untuk jangka waktu tertentu berakibat pada kepemilikan sejumlah sistem dengan *platform* yang berbeda-beda baik dari sisi perangkat keras maupun perangkat lunak serta hanya mampu menunjang area fungsional tertentu.

Kondisi bisnis yang kompetitif saat ini memerlukan SI/TI yang dapat mendukung proses bisnis suatu perusahaan secara menyeluruh, yang bukan terbatas pada fungsi bisnis tertentu saja namun dapat bermanfaat bagi sejumlah area fungsional (*across functional areas*). Kecenderungan ini menuntut pada kebutuhan untuk melakukan integrasi terhadap sistem yang berorientasi fungsi agar dapat sejalan dengan proses bisnis. Kebutuhan akan integrasi perusahaan/*enterprise integration (EI)* dikendalikan oleh sejumlah faktor kunci. Pertama, tekanan lingkungan bisnis yang kompetitif yang mengendalikan para manajemen SI/TI untuk dapat memperpendek siklus hidup pengembangan aplikasi dengan cara guna ulang (*reuse*) layanan aplikasi dan informasi yang telah ada dan bukan dengan menciptakan proses bisnis, layanan aplikasi serta simpanan data yang sama secara berulang-ulang. Kedua, integrasi aplikasi menyediakan manfaat kompetitif bagi perusahaan yang ingin saling berbagi informasi aplikasi baik di dalam perusahaan maupun dengan pihak di luar perusahaan [9].

Berdasarkan faktor-faktor kunci itulah, perusahaan mulai melirik integrasi sebagai solusi atas permasalahan seputar sistem yang mereka miliki. Namun, setiap perusahaan memiliki berbagai kepentingan yang terkait dengan integrasi. Perusahaan yang memerlukan integrasi internal terhadap berbagai sistem yang mendukung area-area fungsional yang berbeda dari suatu bisnis akan melakukan integrasi intra-organisasi secara horizontal. Perusahaan yang memerlukan integrasi antara sistem pada tingkatan kontrol dan manajerial yang berbeda dari suatu organisasi akan melakukan integrasi intra-organisasi secara vertikal. Sedangkan perusahaan yang memerlukan integrasi antar sistem dengan perusahaan lain akan melakukan integrasi inter-organisasi.

Kecenderungan pengembangan sistem yang hanya diperuntukkan bagi fungsi bisnis tertentu juga terjadi di Yayasan Pendidikan “X”. “X” sebagai lembaga yang bergerak dalam bidang pendidikan memiliki sejumlah fungsi bisnis yang ditunjang oleh SI/TI. Namun dalam perkembangannya, pengelolaan SI/TI ini dilakukan secara independen oleh masing-masing unit organisasi. Setiap unit organisasi melakukan pembelian perangkat keras dan pengembangan perangkat lunak pada waktu yang berbeda dan dari vendor yang berbeda. Sebagai contoh adalah aplikasi-aplikasi yang mendukung fungsi akademik dan keuangan yang dibuat oleh sejumlah vendor dan dalam waktu yang berlainan yaitu berkisar dari tahun 1999 sampai dengan tahun 2006. Perbedaan pengembang dan tahun pembuatan jelas berdampak pada kepemilikan sejumlah sistem dengan *platform* yang berbeda-beda,

mulai dari perbedaan teknologi perangkat keras, bahasa pemrograman, sistem pengelola basis data, sistem operasi sampai dengan sistem aplikasi penunjang lainnya. "X" sebagai satu kesatuan perusahaan memerlukan integritas data dan informasi dari seluruh fungsi baik fungsi utama yaitu pendidikan maupun fungsi-fungsi pendukung. Namun integritas data dan informasi ini tidak dapat diperoleh melalui sistem yang dimiliki "X" saat ini karena adanya perbedaan *platform*. Perbedaan *platform* menyebabkan terjadinya "pulau-pulau" informasi karena setiap sistem tidak dapat berkomunikasi untuk saling berbagi pakai data dan informasi.

Enterprise integration merupakan jawaban terhadap permasalahan yang timbul sebagai akibat munculnya "pulau-pulau" informasi. Selama beberapa generasi, pengembangan sistem di "X" diarahkan hanya untuk melayani fungsi bisnis tertentu yang berakibat pada adanya "*stovepipe system*" di dalam perusahaan. Sistem yang dibangun secara *custom built* dengan menggunakan *data storage* serta teknologi pengembangan aplikasi yang tidak standar berdampak luas bagi "X". Sistem menjadi tidak mampu memberikan landasan bagi *business agility* dan tidak mampu menyesuaikan diri terhadap perubahan secara cepat. Penerapan SI/TI ternyata juga tidak dapat memenuhi kebutuhan bisnis (*business requirements*) secara maksimal dan hal ini baru disadari oleh "X" setelah sistem tersebut diimplementasikan dengan munculnya banyak keluhan terhadap ketidakmampuan sistem dalam memenuhi kebutuhan para pelaku organisasi.

Berdasarkan sejumlah permasalahan yang timbul di "X" saat ini sebagai akibat adanya "pulau-pulau" informasi/"*stovepipe system*" dan mengingat pentingnya integrasi sebagai solusi atas permasalahan tersebut, maka pembahasan tesis ini akan fokus pada pembentukan arsitektur integrasi sistem internal yang bersifat *enterprise-wide*. Pengembangan arsitektur akan menghasilkan cetak biru integrasi yang dapat mendukung proses bisnis secara menyeluruh.

2. Tinjauan Pustaka

2.1 Definisi Integrasi Enterprise

Bernstein dan Ruh mendefinisikan *enterprise integration* yaitu :

"Unrestricted sharing of information, services, and business processes among any connected applications or data sources in the enterprise." [4].

Hohpe, Gregor dalam bukunya *Enterprise Integration Patterns* mendefinisikan *enterprise integration* sebagai :

"Enterprise integration has to deal with multiple applications running on multiple platforms in different locations, making the term simple integration pretty much an oxymoron" [5].

Lam, Wing dalam tulisan yang berjudul *Technical Risk Management on Enterprise Integration Projects* mendefinisikan *enterprise integration* sebagai :

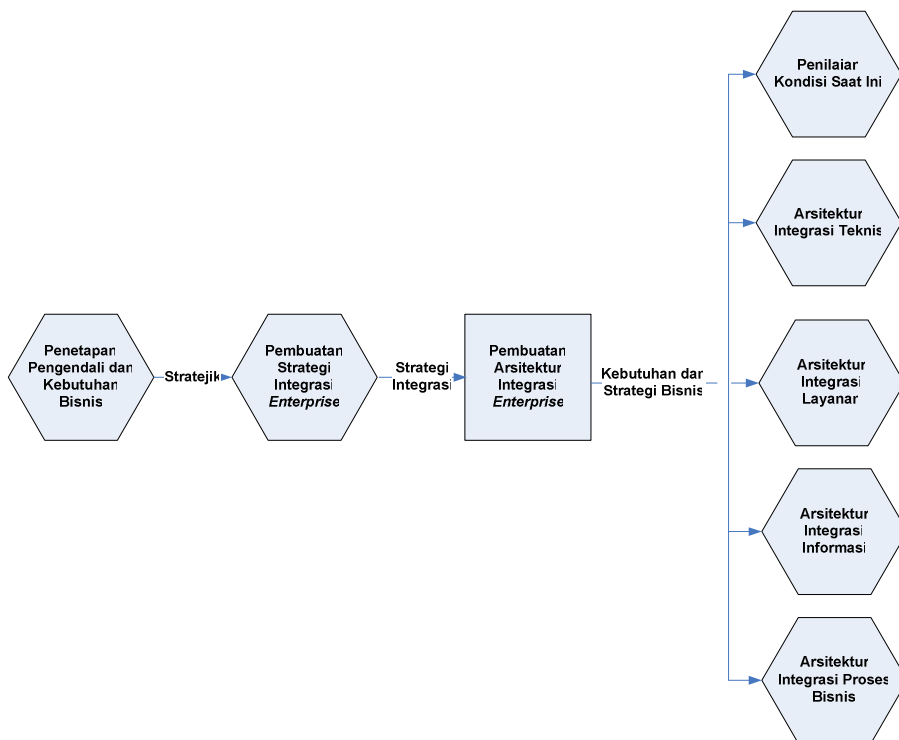
"A general term that refers to the integration of IT systems and business processes both within the enterprise and between different enterprises" [8].

Dari beberapa definisi di atas dapat ditarik kesimpulan bahwa *enterprise integration* (EI) merupakan tugas untuk membuat agar aplikasi-aplikasi yang bekerja pada berbagai *platform* di lokasi yang berbeda dapat bekerja sama guna

menghasilkan suatu kesatuan fungsionalitas, sehingga dapat saling berbagi informasi, layanan dan proses bisnis baik di dalam *enterprise* maupun antar *enterprise*.

2.2 Road Map Integrasi

Road map menjelaskan keseluruhan langkah yang menuntun kegiatan integrasi *enterprise*, secara lengkap dapat dilihat pada Gambar 1. Dari Gambar 1 dapat dilihat bahwa langkah pertama kegiatan integrasi *enterprise* adalah **penetapan pengendali dan kebutuhan bisnis**, langkah ini akan menentukan ruang lingkup integrasi. Langkah berikutnya setelah pendefinisian pengendali dan kebutuhan bisnis adalah **pembuatan strategi integrasi** dan **pembuatan arsitektur integrasi**.



Gambar 1 *Road Map* Integrasi [4]

2.2.1 Pengendali dan Kebutuhan Bisnis

Spesifikasi pengendali dan kebutuhan bisnis merupakan dokumentasi yang menggambarkan apa yang sedang bisnis coba untuk capai [4]. Spesifikasi ini akan menjadi tuntunan bagi proyek dan juga akan digunakan sampai sistem dapat beroperasi untuk menilai dampak pada bisnis.

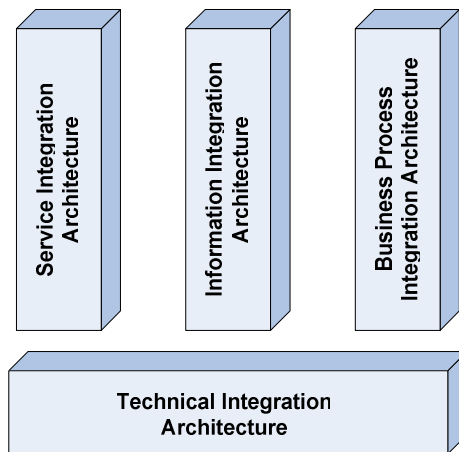
2.2.2 Strategi Integrasi *Enterprise*

Spesifikasi strategi integrasi bisnis merupakan dokumen yang memetakan kebutuhan, strategi dan inisiatif bisnis menjadi strategi dan proyek integrasi. Spesifikasi strategi integrasi dapat dibuat baik pada tingkatan *enterprise* maupun proyek [4].

2.2.3 Arsitektur Integrasi Enterprise

Arsitektur integrasi *enterprise* menyediakan *blueprint* untuk proyek integrasi baik stratejik maupun taktis [4], menggambarkan keseluruhan komponen dari arsitektur. Arsitektur integrasi *enterprise* stratejik meliputi tata kelola untuk memastikan bahwa proyek memenuhi standar-standar yang telah ditetapkan dan terdapat proses untuk pengecualian. Pendekatan-pendekatan taktis untuk mengembangkan infrastruktur teknis ternyata memiliki biaya pemeliharaan yang tinggi dan menghambat *business agility*. Dikarenakan hal tersebut, maka organisasi-organisasi besar dan juga lembaga-lembaga pemerintahan telah menetapkan *framework* arsitektur *enterprise* (*enterprise architecture/EA*). Arsitektur integrasi *enterprise* haruslah cocok dengan seluruh *framework* arsitektur *enterprise*. Prioritas dari pengembangan arsitektur dikendalikan oleh kebutuhan dan strategi bisnis.

Gambar 2 menunjukkan empat domain arsitektur yaitu arsitektur integrasi teknis, arsitektur integrasi layanan, arsitektur integrasi informasi dan arsitektur integrasi proses bisnis.



Gambar 2 Domain Arsitektur Integrasi Enterprise [4]

Penjelasan untuk masing-masing domain arsitektur integrasi *enterprise* adalah :

1. Arsitektur integrasi teknis
Arsitektur integrasi teknis mendefinisikan teknologi untuk seluruh solusi integrasi. Domain ini menjadi dasar guna mendukung komponen arsitektur integrasi *enterprise* yang lain.
2. Arsitektur integrasi layanan
Arsitektur integrasi layanan merupakan *subset* dari arsitektur aplikasi *enterprise*. Didefinisikan sebagai *loosely coupled, reusable business services*, arsitektur aplikasi ini paling fleksibel dan dapat beradaptasi terhadap perubahan bisnis, sehingga memungkinkan integrasi aplikasi secara cepat.
3. Arsitektur integrasi informasi
Arsitektur integrasi informasi menyediakan pandangan secara *enterprise-wide* mengenai data yang terdapat pada sistem yang terpisah. Nilai dari data itu sendiri bergantung pada pemeliharaan integritas data antar sistem. Solusi untuk pemeliharaan nilai, makna dan integritas data antara aplikasi adalah metadata.

Metadata merupakan informasi mengenai data. Semakin deskriptif, akurat dan lengkap metadata, maka akan semakin baik integrasinya. Untuk kepentingan integrasi, metadata dipresentasikan dalam format kanonik sehingga mempermudah pemetaan kembali ke sistem sumber.

4. Arsitektur integrasi proses bisnis

Arsitektur integrasi proses bisnis memodelkan proses bisnis yang melintasi batasan-batasan organisasi. Tujuan dari integrasi adalah untuk meningkatkan proses bisnis dan efisiensi. Arsitektur proses bisnis memaksimalkan *business agility* karena memungkinkan perubahan terhadap proses bisnis diimplementasikan secara cepat.

2.3 Enterprise Architecture Planning

Menurut Spewak, *enterprise architecture planning* (EAP) merupakan “*proses mendefinisikan arsitektur untuk menggunakan informasi guna mendukung bisnis dan rencana untuk mengimplementasikan arsitektur tersebut.*”[11]

EAP merupakan proses untuk mendefinisikan kedua *top layer* dari *framework* arsitektur sistem informasi Zachman. EAP menghasilkan *blueprint* mengenai data, aplikasi dan teknologi yang menghasilkan solusi jangka panjang yang *cost-effective*, bukan hanya perbaikan secara cepat. Gambar 3 menunjukkan 7 komponen atau fase EAP, yang menjelaskan bagaimana mendefinisikan arsitektur dan perencanaan. Komponen-komponen tersebut terbentuk sebagai *layer*, dimana tiap *layer* merepresentasikan fokus tugas yang berbeda, yaitu :

1. Layer 1-Where We Start

Planning initiation. Memulai EAP pada jalur yang benar, termasuk menentukan metodologi yang digunakan, siapa yang harus dilibatkan dan *toolset* apa yang digunakan.

2. Layer 2-Where We Are Today

- a. *Business modeling.* Menyusun *knowledge base* mengenai bisnis dan informasi yang digunakan untuk melaksanakan bisnis.
- b. *Current systems & technology.* Mendefinisikan sistem aplikasi apa yang terdapat saat ini dan *platform* teknologi yang mendukung.

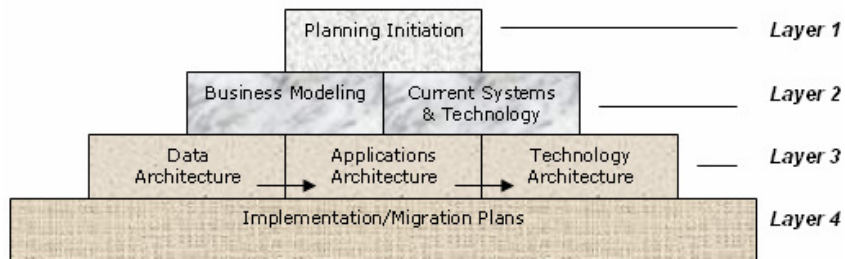
3. Layer 3-Where We Want to Be in the Future

- a. *Data architecture.* Mendefinisikan data yang diperlukan untuk mendukung bisnis.
- b. *Application architecture.* Mendefinisikan aplikasi yang diperlukan untuk mengelola data dan mendukung fungsi-fungsi bisnis.
- c. *Technology architecture.* Mendefinisikan *platform* teknologi yang diperlukan untuk menyediakan lingkungan bagi aplikasi yang mengelola data dan mendukung fungsi-fungsi bisnis.

Panah pada *layer* ini menunjukkan bahwa *data architecture* didefinisikan terlebih dahulu, lalu berturut-turut mendefinisikan *application architecture* dan *technology architecture*. Hal ini berbeda dengan metoda perencanaan sistem tradisional yang melakukan sebaliknya, dimana pertama-tama menentukan *hardware*, kemudian aplikasi yang berjalan pada *hardware*, dan terakhir data yang perlu diproses.

4. Layer 4-How We Get There

Implementation/migration plans. Mendefinisikan urutan langkah untuk mengimplementasikan aplikasi, jadwal implementasi, analisis manfaat/biaya dan mengajukan jalur yang jelas untuk melakukan migrasi dari *where we are today* ke *where we want to be*.



Gambar 3 Komponen *Enterprise Architecture Planning* [11]

2.4 Value Chain

Kunci analisis *value chain* adalah memahami aktivitas di dalam institusi yang menciptakan manfaat kompetitif serta pengaturan aktivitas tersebut lebih baik dari institusi lain pada industri. Porter (1985) mengemukakan bahwa aktivitas bisnis dapat dikelompokkan menjadi dua :

1. *Primary activities*, yang secara langsung berkaitan dengan produksi dan pengiriman produk atau layanan; serta
2. *Support activities*, yang mendukung *primary activities*, tidak terlibat langsung dalam produksi, namun memiliki potensi meningkatkan efisiensi dan efektivitas.

2.10 Four Stage Life Cycle Business System Planning (BSP)

Four stage life cycle [7] adalah *tool* yang digunakan untuk menemukan turunan dari fungsi bisnis yang terkait dengan produk atau layanan yang diberikan oleh fungsi bisnis tersebut. *Four stage life cycle* pada BSP digunakan pada tahap pendefinisian proses bisnis. Keempat siklus yang digunakan, yaitu :

1. Tahap I, *Requirement, Planning, Measurement and Control*.
Aktivitas yang menentukan berapa banyak produk atau layanan yang dibutuhkan, rencana untuk mendapatkannya dan pengukuran serta kontrol yang terkait dengan rencana.
2. Tahap II, *Acquisition*.
Aktivitas yang dilakukan untuk mengembangkan produk atau layanan atau untuk mendapatkan sumber daya yang akan dipergunakan untuk kegiatan pengembangan.
3. Tahap III, *Stewardships*.
Aktivitas untuk membentuk, mempertajam, memodifikasi atau merawat dukungan sumber daya dan untuk menyimpan atau menelusuri produk atau layanan.
4. Tahap IV, *Retirement/Disposition*.
Aktivitas dan keputusan yang mengakhiri tanggung jawab organisasi terhadap suatu produk/layanan atau isyarat terhadap berakhirnya penggunaan suatu sumber daya.

3. Analisis Kondisi *Enterprise*

3.1 Analisis Kondisi Saat Ini (*As-is*)

Analisis kondisi *enterprise* saat ini (*as-is*) bertujuan untuk melihat kondisi “X” saat ini dengan pendekatan EAP yang terdiri atas tahapan inisiasi perencanaan, pemodelan bisnis serta penilaian kondisi sistem dan teknologi.

3.1.1 Inisiasi Perencanaan

Inisiasi perencanaan dilakukan agar proyek dapat diproses secara cepat dalam arahan yang benar semenjak awal. Pendekatan *Enterprise Architecture Planning* (EAP) dalam studi kasus ini digunakan untuk memberikan landasan dalam mengatasi berbagai permasalahan terpisahnya aplikasi *legacy* sehingga timbulnya “pulau-pulau” data yang berdampak pada kurang optimalnya dukungan sistem informasi terhadap bisnis serta menjadi arahan bagi pengembangan sistem.

Langkah-langkah yang dilakukan pada fase inisiasi perencanaan adalah :

1. Menentukan ruang lingkup dan sasaran perencanaan arsitektur *enterprise*.

Berdasarkan analisa terhadap profil dan kegiatan “X”, dapat ditarik kesimpulan bahwa fungsi bisnis utama “X” adalah pendidikan, baik pendidikan non formal maupun formal. Oleh karenanya, pembentukan arsitektur integrasi didasarkan pada kebutuhan fungsi bisnis utama yaitu pendidikan dan fungsi-fungsi pendukungnya. Area-area yang akan dikaji, yang kemudian akan menjadi ruang lingkup dalam pembentukan arsitektur integrasi adalah area penerimaan siswa/mahasiswa baru, area pengelolaan kegiatan akademik, area pengelolaan wisuda, alumni dan bursa kerja serta area pengelolaan pembayaran biaya pendidikan. Sasaran dari pembentukan arsitektur integrasi adalah menyediakan artifak-artifak berupa daftar fungsi/proses bisnis, unit organisasi, entitas data, aplikasi dan landasan teknologi yang dapat dijadikan dasar pengembangan sistem informasi terintegrasi.

2. Menentukan visi.

Visi SI/TI “X” disesuaikan dengan visi organisasi “X”.

3. Menentukan metodologi.

Metodologi yang digunakan untuk pembentukan arsitektur integrasi adalah :

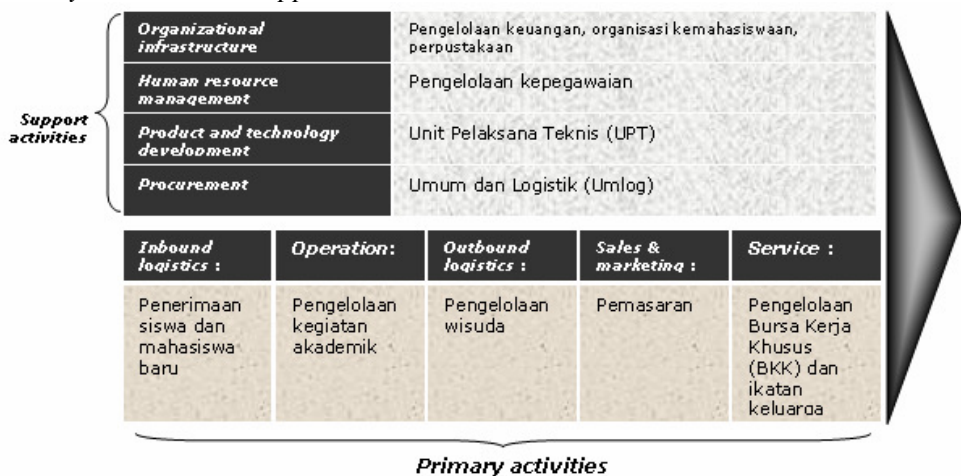
- a. Inisiasi perencanaan (*planning initiation*).
- b. Pemodelan bisnis (*business modeling*).
- c. Analisis arsitektur sistem dan teknologi saat ini (*current systems and technology architecture*).
- d. Arsitektur data, aplikasi dan teknologi (*data architecture, application architecture, technology architecture*).
- e. Pengembangan arsitektur integrasi.

3.1.2 Pemodelan Bisnis

Fungsi merupakan sekumpulan aktivitas yang dilakukan dalam bisnis dan fungsi didefinisikan berdasarkan bagian-bagiannya. Definisi fungsi bisnis hanyalah didasarkan pada aksi-aksi yang dilakukan, bukan pada organisasinya maupun orang yang bertanggung jawab untuk melaksanakan suatu fungsi. Untuk mengidentifikasi dan mendefinisikan fungsi-fungsi bisnis yang terdapat di “X”, langkah-langkah yang harus dilakukan adalah :

1. Mendefinisikan area-area fungsional utama menggunakan konsep “*value-added*” Michael Porter.
2. Memecah/mendekomposisi tiap area fungsional menjadi sub fungsi sampai fungsi tersebut menjadi *single-action*, dapat dilakukan secara berulang, memiliki *outcome* dan dapat dikaitkan dengan unit organisasi tertentu menggunakan *Four Stage Life Cycle Business System Planning*.
3. Membuat relasi antara fungsi-fungsi yang telah terinci dengan unit-unit organisasi yang melaksanakannya dalam bentuk matriks.

Pendefinisian aktivitas area-area fungsional utama di “X” menggunakan rantai nilai (*value chain*) Michael Porter, seperti yang terdapat pada Gambar 4. Dalam Gambar 4 tersebut, fungsi-fungsi bisnis di “X” dikelompokkan menjadi 2 yaitu *primary activities* dan *support activities*.



Gambar 4 Rantai Nilai “X”

3.1.3 Arsitektur Sistem dan Teknologi Saat Ini

Tujuan dari tahapan ini adalah untuk mendokumentasikan dan mendefinisikan seluruh *platform* sistem dan teknologi yang dimiliki, dikelola serta digunakan *enterprise* saat ini. *Deliverable* dari tahapan ini adalah *Information Resource Catalog* (IRC), disebut juga *System Encyclopedia* atau *System Inventory*. Langkah-langkah untuk membangun IRC adalah :

1. Mempersiapkan koleksi data aplikasi dan teknologi.
2. Mengumpulkan data IRC.

Tujuan dari tahapan ini adalah untuk menentukan macam-macam data yang disertakan dalam IRC. Langkah-langkah dalam pengumpulan data yaitu menentukan data mengenai aplikasi dan mengidentifikasi *platform* teknologi. Dokumentasi aplikasi bertujuan mengidentifikasi aplikasi apa saja yang telah dimiliki, dikelola serta digunakan oleh masing-masing unit organisasi di “X”. Saat ini data yang dihasilkan oleh proses bisnis di “X” disimpan dalam basis data aplikasi-aplikasi yang berbeda dan tidak terintegrasi. Fungsi bisnis yang telah didukung oleh aplikasi adalah fungsi akademik dan keuangan. Aplikasi-aplikasi tersebut dianggap telah mampu mendukung suatu fungsi bisnis tertentu namun

tidak dapat saling mendukung fungsi bisnis lain, karena tidak terhubung satu sama lain dan memiliki *platform* yang berbeda.

Identifikasi *platform* teknologi merupakan definisi dekomposisi secara hirarkis mengenai jenis-jenis *platform* teknologi yang terdapat dalam suatu *enterprise*. *Platform* teknologi di “X” terbagi ke dalam 3 kelompok besar yaitu perangkat keras (*hardware*), perangkat lunak (*software*) dan perangkat komunikasi (*communication*).

3.1.4 Hasil Analisis Kondisi “X” Saat Ini

Core business “X” adalah pendidikan, hal ini dapat dilihat pada rantai nilai “X” dengan menggunakan rantai nilai Porter (Gambar 4). Dalam melaksanakan aktivitas-aktivitas utamanya, “X” melakukan pemisahan pengelolaan antara kegiatan akademik untuk pendidikan non formal dengan pendidikan formal.

Pemisahan pengelolaan dari sisi bisnis, berdampak pada pengelolaan sistem informasi/teknologi informasi (SI/TI), dimana mulai dari pengadaan, pengoperasian sampai dengan pemeliharaan SI/TI dilakukan secara independen oleh masing-masing bagian pada unit-unit organisasi untuk memenuhi kebutuhan suatu fungsi bisnis yang mendesak saat itu (*temporer*). Pengelolaan SI/TI yang dilakukan secara independen ini menyebabkan perbedaan pada spesifikasi perangkat keras dan *platform* perangkat lunak di masing-masing bagian yang mengelola fungsi bisnis.

Berdasarkan hasil analisis dan informasi mengenai aplikasi pada IRC, terdapat 4 kelompok aplikasi yang masing-masing adalah aplikasi untuk mendukung fungsi akademik dan aktivitas pendukung keuangan untuk dua unit organisasi yang berbeda yaitu “A” dan “B”. Aplikasi-aplikasi tersebut dibuat oleh beberapa vendor dalam waktu yang berbeda sehingga tidak mengherankan jika aplikasi-aplikasi tersebut dibuat dalam berbagai teknologi bahasa pemrograman dan *database management system* (DBMS) yang berbeda-beda.

Perbedaan bahasa pemrograman dan DBMS atau perbedaan *platform* dan fungsionalitas aplikasi menjadikan aplikasi-aplikasi berdiri sendiri-sendiri (*stovepipe*) untuk melayani suatu fungsi bisnis akademik dan juga keuangan pada satu unit organisasi dan tidak dapat saling mempertukarkan data serta fungsionalitas antar fungsi dan juga antar unit organisasi sebagai satu kesatuan (*enterprise-wide*).

3.2 Menentukan Kebutuhan Arsitektur Mendatang (*To-be*)

Menurut Spewak, tahapan-tahapan perencanaan arsitektur *enterprise* dikelompokkan ke dalam 4 layer yaitu layer 1 (*where we start*), layer 2 (*where we are today*), layer 3 (*where we want to be in the future*) dan layer 4 (*how we get there*). Pada bagian sebelumnya telah dilakukan analisis terhadap kondisi sistem informasi “X” saat ini (*as is*). Tahapan selanjutnya adalah menentukan kebutuhan sistem informasi “X” di masa mendatang (*to be*).

3.2.1 Arsitektur Data

Arsitektur data merupakan arsitektur pertama dari 3 arsitektur yang harus didefinisikan karena kualitas data merupakan produk dasar dalam fungsi sistem

informasi. Arsitektur data berisi entitas-entitas data dimana masing-masing entitas tersebut memiliki atribut dan membentuk relasi dengan entitas data lain.

Langkah-langkah dalam membentuk arsitektur data adalah :

1. Mendaftar kandidat entitas-entitas data.
2. Mendefinisikan entitas, atribut dan relasi.
3. Merelasikan entitas dengan fungsi bisnis.

3.2.2 Arsitektur Aplikasi

Arsitektur aplikasi merupakan definisi mengenai apa yang harus dilakukan aplikasi untuk mengelola data dan menyediakan informasi bagi pelaksana fungsi-fungsi bisnis. Tahapan-tahapan untuk menghasilkan arsitektur aplikasi adalah :

1. Mendaftar kandidat aplikasi.
2. Mendefinisikan aplikasi.
3. Merelasikan aplikasi dengan fungsi.

3.2.3 Arsitektur Teknologi

Arsitektur teknologi dibuat untuk mendefinisikan teknologi yang diperlukan untuk dapat menyediakan lingkungan bagi aplikasi dalam pengelolaan data. Sama dengan arsitektur data dan aplikasi, arsitektur teknologi juga merupakan model konseptual yang mendefinisikan *platform*. Tahapan-tahapan dalam pembentukan arsitektur teknologi adalah :

1. Mengidentifikasi prinsip dan *platform* teknologi.
2. Mendefinisikan *platform*.
3. Merelasikan *platform* teknologi dengan fungsi-fungsi bisnis.
4. Merelasikan *platform* teknologi dengan aplikasi.

3.3 Pengendali dan Requirement Bisnis untuk Integrasi Enterprise

Terdapat sejumlah *business initiative* yang mengendalikan *requirement* integrasi, diantaranya adalah untuk mengurangi *business cycle time* sehingga dapat meningkatkan efisiensi dan daya saing, meningkatkan kepuasan konsumen, *merger* dan akuisisi serta untuk mematuhi suatu regulasi. Adanya perbedaan *requirement* bisnis akan berdampak pada teknologi integrasi yang digunakan.

4. Pembentukan Arsitektur Integrasi

4.1 Mengevaluasi Gap Diantara Kondisi As-is dan To-be

Pada bagian sebelumnya telah dianalisis kondisi *enterprise* saat ini (*as-is*) serta kebutuhan sistem mendatang (*to-be*). Penilaian terhadap kondisi saat ini menunjukkan kapabilitas sistem yang sedang berjalan dan tentu saja terdapat kesenjangan (*gap*) diantara kondisi saat ini dengan kebutuhan untuk dapat mencapai kondisi ideal. Melalui hasil evaluasi kesenjangan inilah nantinya akan dibuat kebijakan penyelesaian permasalahan integrasi.

4.1.1 Perbandingan Data

Entitas data yang dihasilkan dan digunakan pada sistem saat ini adalah pada fungsi penerimaan siswa/mahasiswa baru, kegiatan akademik, pengelolaan wisuda dan pembayaran biaya pendidikan. Aliran data yang disimbolkan dengan garis

beranak panah menandakan bahwa suatu area sistem menggunakan data pada area sistem lainnya, sebagai contoh area sistem kedua yaitu akademik memiliki aliran data dari area sistem pertama yaitu penerimaan siswa/mahasiswa baru. Hal tersebut berarti bahwa sistem akademik menggunakan data pendaftar, jadwal usm dan hasil usm yang diciptakan pada sistem penerimaan siswa/mahasiswa baru.

Melalui pemetaan yang telah dilakukan terhadap data pada aplikasi *legacy* dengan arsitektur ideal, ditemukan bahwa terdapat 4 entitas data dari keseluruhan 30 entitas data ideal atau 13.33 % yang belum tersedia pada aplikasi *legacy*. Jadi 86.67 % entitas data ideal sebenarnya telah dihasilkan dari aplikasi *legacy*.

Permasalahan lain dari hasil pemetaan adalah adanya 5 atau 16.67 % entitas data acuan (master) yang dihasilkan secara berulang oleh sejumlah aplikasi *legacy* yaitu entitas pendaftar, mata kuliah, siswa, mahasiswa dan dosen. Sedangkan 83.33 % entitas-entitas lainnya dikelola secara mandiri oleh unit-unit organisasi sehingga memiliki beragam format dan tidak terintegrasi. Hal ini tentu saja merepotkan pengelolaan sistem karena seharusnya isi data yang sama dibuat berulang-ulang (terjadi redundansi). Berdasarkan hal inilah maka diperlukan integrasi data yang dikelola oleh aplikasi-aplikasi *legacy*.

4.1.2 Perbandingan Aplikasi

Terdapat 4 aplikasi dari total 29 aplikasi atau 13.79 % yang termasuk ke dalam pengembangan baru, yaitu aplikasi yang berhubungan dengan aplikasi promosi dan pengelolaan BKK. Sedangkan aplikasi lain sebesar 86.21 % dipertahankan atau dimodifikasi dari aplikasi lama (*legacy*) dengan melakukan integrasi.

4.1.3 Perbandingan Teknologi

Perbandingan dokumentasi teknologi yang ada dan digunakan saat ini dengan arsitektur teknologi ideal, diperoleh beberapa kesimpulan sebagai berikut :

1. Teknologi jaringan lokal (LAN) saat ini masih dapat digunakan untuk mendukung aplikasi dan data, namun perlu dipertimbangkan untuk meningkatkan konfigurasi jaringan sehingga dapat mendukung aplikasi berbasis internet. Saat ini teknologi jaringan internet telah tersedia namun hanya digunakan untuk kegiatan belajar mengajar dan belum dimanfaatkan untuk mendukung fungsi bisnis.
2. Distribusi data dan file saat ini sebagian besar terpusat di *server*, hal ini sangat membebani kerja *server*. Oleh karenanya perlu adanya pemisahan fungsi *server* sebagai penyedia layanan jaringan dengan penyedia data dan aplikasi.
3. Diperlukan *middleware* yang dapat mengkomunikasikan data dari basis data berbasis bahasa *Clipper* ke basis data SQL Server dan begitu juga sebaliknya.
4. Setiap aplikasi tidak menyediakan *interface* agar dapat berkomunikasi dengan aplikasi lain sehingga tidak dapat dilakukan integrasi antar aplikasi pada level *interface*.

4.2 Pembentukan Arsitektur Integrasi Teknis

Arsitektur integrasi teknis menyajikan *building code* bagi proyek integrasi karena berisi spesifikasi yang akan diacu oleh proyek saat memilih teknologi integrasi. Arsitektur ini terdiri atas tuntunan dan batasan rancangan mengenai

bagaimana seharusnya aplikasi dikembangkan. Oleh karenanya, spesifikasi harus dapat mendefinisikan seluruh aspek arsitektur integrasi dan mudah diakses sehingga informasi dapat mudah ditemukan dan dipahami. Arsitektur teknis haruslah dikendalikan oleh *business requirements* dan mampu memenuhi kebutuhan di masa mendatang.

Berdasarkan hasil analisa terhadap proses bisnis, kondisi sistem dan teknologi saat ini, maka kebutuhan "X" adalah melakukan integrasi terhadap aplikasi yang ada (*legacy*), integrasi data dari berbagai unit organisasi yang akan menghasilkan informasi terintegrasi, serta integrasi gabungan dengan aplikasi baru yang akan dikembangkan. Integrasi aplikasi menjadi solusi bagi permasalahan di "X" karena adanya kebutuhan untuk tetap dapat menggunakan basis data dan layanan pada aplikasi yang ada (*legacy*) semaksimal mungkin, sehingga setiap unit organisasi dapat saling berbagi data dan proses tanpa membuat perubahan terhadap aplikasi maupun struktur data secara terus-menerus untuk mengikuti kebutuhan bisnis.

Berdasarkan kebutuhan untuk menggabungkan aplikasi yang tidak merubah logik aplikasi serta memperhatikan skema relasi basis data dan skema aplikasi yang memerlukan pertukaran data antar aplikasi dari tabel-tabel master yang tidak memiliki keterkaitan logik secara erat antar basis data maka disimpulkan bahwa keempat kelompok aplikasi memerlukan adanya integrasi pada tingkatan data.

Integrasi pada level data dipilih karena tidak adanya akses terhadap logik atau *source code* dari masing-masing aplikasi. Integrasi dapat dilakukan pada level data dengan menempatkan *software* diantara basis data dari keempat aplikasi. *Software* ini bertugas untuk melakukan ekstraksi informasi dari suatu basis data, melakukan *reformat* terhadap data dengan merubah isi dan skema jika diperlukan serta melakukan *update* basis data. Data direplikasi diantara basis data saat terjadi *update* dari sisi basis data manapun ke tabel yang bersesuaian.

4.3 Pembentukan Arsitektur Integrasi Layanan

Arsitektur integrasi layanan mendefinisikan aplikasi bisnis sebagai komponen fungsionalitas bisnis yang dapat diguna ulang dan mudah dirubah serta bagaimana komponen-komponen tersebut saling terkait. Konsep ini merupakan konsep arsitektur yang berbasis layanan (*service-oriented architecture/SOA*). Tahapan-tahapan dalam membuat arsitektur integrasi layanan adalah :

1. Menentukan *business events*.
2. Menentukan layanan.

4.4 Pembentukan Arsitektur Integrasi Informasi

Informasi dan data merupakan hal yang penting dalam proyek integrasi karena permasalahan utama pada seluruh proyek integrasi adalah bagaimana memungkinkan *interoperability* diantara sistem yang memiliki data dalam berbagai struktur dan format. Arsitektur integrasi informasi mendefinisikan infrastruktur dan proses yang memungkinkan informasi diakses pada sistem.

Dalam integrasi, aliran informasi dan juga data perlu digambarkan untuk memperoleh kejelasan mengenai data dan informasi apa saja yang sebenarnya dihasilkan atau diperlukan dari atau ke sistem. Penggambaran aliran informasi dalam sistem tersebut akan menggunakan *data flow diagram* (DFD).

4.5 Pembentukan Arsitektur Integrasi Proses Bisnis

Tujuan integrasi adalah untuk mendukung peningkatan proses bisnis dalam rangka meningkatkan efisiensi bisnis. Integrasi level proses mendefinisikan interaksi diantara sistem melalui definisi *business workflow*. Peran arsitektur integrasi proses adalah untuk menciptakan model dan definisi proses sebagai entitas yang dikelola sehingga mudah beradaptasi terhadap perubahan bisnis. Arsitektur integrasi proses mendefinisikan *end-to-end business process* yang diotomatisasi pada sistem dengan *platform* yang berbeda-beda.

4.6 Strategi Integrasi

Arsitektur integrasi yang telah dibangun merupakan *blueprint* integrasi teknologi, layanan, informasi dan proses bisnis yang menjadi dasar bagi pengembangan dan pengelolaan sistem informasi sehingga selaras dengan bisnis *enterprise*. Arsitektur integrasi dibangun dengan didasarkan pada dorongan bisnis kemudian pada kebutuhan data, aplikasi dan teknologi. Untuk menerapkan hasil pengembangan arsitektur, maka diperlukan strategi sehingga dapat menerapkan integrasi. Hasil dari penerapan strategi integrasi diharapkan menjadi acuan dalam implementasi kegiatan integrasi.

5. Kesimpulan

Berdasarkan hasil pembahasan mengenai pembentukan integrasi *enterprise* dengan studi kasus Yayasan Pendidikan “X”, maka diperoleh beberapa kesimpulan sebagai berikut :

1. Berdasarkan hasil analisis kondisi *as-is* yang diterapkan pada “X” menunjukkan bahwa kepemilikan sistem informasi saat ini tidak dilandaskan pada pemahaman terhadap fungsi-fungsi bisnis secara menyeluruh sehingga muncul “pulau-pulau” informasi. “Pulau-pulau” informasi tersebut ditunjukkan dengan didapatinya 83.33 % data yang dikelola secara independen yang berdampak pada isolasi data dan menghasilkan 16.67 % redundansi data.
2. Kebijakan SI/TI yang dibuat oleh “X” juga hanya fokus pada pemenuhan kebutuhan yang bersifat temporer sehingga mewarisi berbagai teknologi *obsolete* yang tidak mampu beradaptasi dengan teknologi baru. Permasalahan terjadi ketika adanya kebutuhan untuk mengintegrasikan data dan informasi, namun tidak dapat dipenuhi karena ternyata setiap sistem berbeda *platform*.
3. Berdasarkan hasil penentuan kebutuhan data, aplikasi dan teknologi ideal (*to-be*) menunjukkan bahwa teridentifikasi 30 entitas data dan 29 aplikasi yang harus tersedia di “X”. Sebenarnya data dan fungsionalitas sebagian besar telah tersedia pada aplikasi *legacy* namun data dan aplikasi tersebut redundan, tidak konsisten dan terisolasi pada unit-unit organisasi “A” dan “B”.
4. Kesenjangan antara kondisi *as-is* dan *to-be* disertai dengan pengendali dan *requirement* bisnis menunjukkan bahwa “X” memerlukan integrasi yang tidak hanya melandaskan pada penggunaan teknologi tertentu namun memerlukan hasil integrasi yang stabil terhadap perubahan *trend* teknologi. Hal ini diatasi dengan pembentukan arsitektur integrasi teknis, layanan, informasi dan proses yang bersifat strategik.
5. Integrasi *enterprise* bagi “X” telah terbukti dapat memperpendek siklus hidup pengembangan sistem karena dapat melakukan guna ulang (*reuse*) layanan

aplikasi dan informasi yang telah ada dan tidak perlu membuat proses bisnis, layanan aplikasi serta simpanan data yang sama secara berulang-ulang. Hal ini ditunjukkan melalui pembahasan integrasi keempat kelompok aplikasi *legacy* di "X" dengan cara melakukan translasi dan transformasi data antar simpanan data tanpa perlu membuat basis data baru.

6. Integrasi *enterprise* mampu mengkomunikasikan sistem berbeda *platform* di "X" sehingga dapat mengatasi permasalahan adanya pulau-pulau informasi sebagai akibat isolasi pengelolaan data. Hal ini ditunjukkan melalui pembahasan pembentukan arsitektur integrasi teknis, layanan, informasi dan proses yang memberikan kemampuan integrasi sistem yang tidak tergantung pada teknologi tertentu.
7. Kegiatan integrasi yang selama ini dilakukan oleh banyak perusahaan seringkali hanya memandang bahwa integrasi hanya sekedar penggunaan teknologi dan infrastruktur. Berdasarkan pembahasan tesis, ditunjukkan bahwa ketergantungan pada teknologi tertentu untuk kegiatan integrasi membuat sistem yang dihasilkan tidak mampu memberi landasaan bagi *business agility*. Integrasi *enterprise* harus dimulai dengan pemahaman terhadap permasalahan bisnis dan mengetahui bagaimana bisnis berhubungan dengan konsumen dan rekanan bisnisnya.

Daftar Pustaka

- [1] Beynon, Paul-Davies, *Database Systems*, third edition, Palgrave MacMillan.
- [2] Cummins, Fred A. (2002), *Enterprise Integration: An Architecture for Enterprise Application and Systems Integration*, Wiley, USA.
- [3] Erl, Thomas (2004), *Service Oriented Architecture : A Field Guide to Integrating XML and Web Services*, Pearson Education, Inc., USA.
- [4] Gold-Bernstein, Beth (2005), *Enterprise Integration : The Essential Guide to Integration Solutions*, Pearson Education, Inc., USA.
- [5] Hohpe, Gregor (2003), *Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions*, Addison Wesley.
- [6] Husein, Inne Gartina (2004), *Model Enterprise Application Integration Sistem Manajemen Billing di PT TELKOM Divisi Regional III*, Tesis Program Magister, Institut Teknologi Bandung.
- [7] International Business Machine (IBM) Corporation (July 1981), *Business Systems Planning*, 3rd edition.
- [8] Lam, Wing (2004), *Technical Risk Management on Enterprise Integration Projects*, vol. 13, Communications of the Association for Information Systems.
- [9] Linthicum, David S. (2000), *Enterprise Application Integration*, Addison Wesley Longman.
- [10] Mitchell, Victoria L. (December 2006), *Knowledge Integration and Information Technology Project Performance*, vol.30 no.4, pp.919-939, MIS Quarterly.
- [11] Spewak, Steven H. (1992), *Enterprise Architecture Planning : Developing a Blueprint for Data, Applications and Technology*, John Wiley & Sons, Inc.
- [12] Triloka, Joko (2007), *Pemodelan Arsitektur Enterprise untuk Mendukung Sistem Informasi Terintegrasi di Bidang Akademik Menggunakan Enterprise*

- Architecture Planning (Studi Kasus : Universitas Islam Negeri Sunan Kalijaga Yogyakarta)*, Tesis Program Magister, Institut Teknologi Bandung.
- [13] Ward, John & Peppard, Joe (2002), *Strategic Planning for Information Systems*, 3rd edition, John Wiley & Sons, Ltd.
 - [14] www.3ht.com, diakses tanggal 31-01-2008 jam 15.39.
 - [15] www.bea.com, diakses tanggal 22-09-2007 jam 11.49.
 - [16] www.dciexpo.com, diakses tanggal 31-01-2008 jam 14.48.
 - [17] www.ilmukomputer.com, diakses tanggal 25-09-2007 jam 17.40.
 - [18] www.javaworld.com, diakses tanggal 22-09-2007 jam 12.05.
 - [19] www.zifa.com, diakses tanggal 27-04-2007 jam 10.04.

Representasi Grafik Multi-Level Berbasis SVG untuk Aplikasi Rich-Content Majalah Mobile

¹⁾ Adi Nugroho, ²⁾ Theophilus Erman Wellem, ³⁾ Geuis Puspita Dewi

Fakultas Teknologi Informasi

Universitas Kristen Satya Wacana

Jl. Diponegoro 52 – 60, Salatiga 50711, Indonesia

Email : ¹⁾ cmbeling@gmail.com, ²⁾ erman_wellem@yahoo.com,
³⁾ geuis_puspita@yahoo.com

Abstract

The discovery and creation of a new variety of mobile content is in a huge demand, but its development is still limited due to the limitations of mobile devices, especially in informational contents; most of them are still text-based, for example SMS-based news and RSS Feeds. This is because handling graphical data is still expensive regarding resources of the mobile device, which means the rendering and parsing process will suffer.

This study investigates how Scalable Vector Graphics (SVG), an open standard XML-based graphic format, can be used for representing graphical information hierarchically in a multi-level representation system. Based on this system, a new Java-based mobile-magazine content service, which contains richer information as a real magazine, is developed and tested for its performances.

From the experiments and testing, the application can pass the Sun Microsystems's Unified Testing Criteria for Java (TM) Technology-based Applications for Mobile Devices Version 2.1 at the level of 93.75% and shows a good performance of parsing and rendering of the graphical information and image.

Keywords : mobile computing, multi-level graphics representation, mobile magazines, mobile contents

1. Pendahuluan

Teknologi komputasi *mobile* dan komunikasi *wireless* telah mengubah praktek bisnis dan *enterprise* tradisional. Berbagai model perangkat, pekerjaan dan aspek-aspek lain yang bersifat *mobile* semakin dikembangkan baik oleh peneliti di bidang industri maupun akademis karena diharapkan akan memberi paradigma baru dalam praktek bisnis maupun kehidupan yang lebih baik. Di sisi perangkat keras (*hardware*), pengembangan perangkat *mobile* sejenis telepon selular dan *smart-phone* saat ini telah mendukung *rich multimedia content*. Hal ini juga didukung dengan adanya sistem komunikasi generasi ketiga yaitu *3G* yang memungkinkan akses jaringan yang lebih baik.

Dari sisi bisnis, perkembangan di bidang perangkat keras (*hardware*) dan jaringan saja tidak cukup. Sebagai contoh, peluncuran teknologi *3G* di beberapa negara di Asia menunjukkan respon pengguna jasa selular ternyata tidak sebaik yang diharapkan pada awalnya. Kurangnya *content* yang ditawarkan oleh berbagai pihak yang berkecimpung dalam dunia telekomunikasi disebut sebagai salah satu faktor utama kegagalan teknologi baru ini. Tanpa adanya *content*, teknologi *3G* akan menjadi percuma [1]. Selain itu, bisnis layanan *content* diyakini menjadi

pemicu pertumbuhan ekonomi di era *digital*. Kenyataannya, pendapatan penyedia jasa selular dari penjualan *content* memang besar [2].

Content yang telah dikenal dalam seluler generasi kedua (*GSM*) diantaranya adalah dari aplikasi *SMS* (berbasis teks) terdapat beberapa *content* yang dikembangkan, seperti kuis, horoskop, berita (*news*), dan lain-lain. Sedangkan dari aplikasi berbasis multimedia berupa nada dering (*ring tone*), nada tunggu (*ring back tone*), atau animasi *wallpaper*. Umumnya operator tidak membuat sendiri *content* yang akan ditawarkan kepada konsumen selular. Peran tersebut diserahkan kepada penyedia layanan *content* (*content provider*).

Sebagai studi kasus pada penelitian ini akan dicoba untuk merancang suatu bentuk aplikasi layanan *content* informasi yang baru berupa majalah *mobile*, yaitu sebuah *content* majalah yang diadaptasi untuk perangkat selular. Karakteristik majalah berbeda dari media informasi cetak yang paling memasyarakat yaitu koran. Mereka terbit menurut edisi, bisa setiap minggu atau setiap bulan, lebih kaya gambar, warna dan desain, serta sering berupa kumpulan artikel yang informasinya tidak mudah kadaluwarsa. Namun ini juga sekaligus daya tarik majalah, pembacanya seringkali menyimpan atau mengoleksi majalah dalam waktu lama untuk dibaca kembali. Mengadaptasi karakteristik majalah ini ke dalam bentuk *mobile service* di lingkungan *mobile* tidak mudah direalisasikan karena keterbatasan perangkat dan adanya gap antara penyedia *content* dan pelaku bisnis selular. Operator selular tidak punya *resource* dan *content* yang dibutuhkan, sebaliknya pihak penerbit majalah tidak memiliki akses ke pasar selular. Karena itu, adanya sarana yang dapat menghubungkan kerja sama kedua pihak ini akan menjadi faktor kesuksesan yang bersifat kritical. Di Eropa, *Mobile Magazine* merupakan bentuk baru *mobile service* yang sedang dikembangkan dan dipercaya mempunyai prospek bisnis masa depan yang bagus. Proyek ini disebutkan masih merupakan pengembangan *prototype* infrastruktur layanan secara global [9].

Fokus utama penelitian ini merupakan langkah awal dari realisasi dari sebuah arsitektur *mobile magazine services*, yaitu pada perancangan format *content* yang dapat digunakan dan sekaligus aplikasi *client* yang akan menggunakannya. Untuk mendukung simulasi pada perancangan aplikasi, dibuat sebuah *server* yang memungkinkan aplikasi untuk melakukan *download content* secara aktual. Untuk menyesuaikan dengan kemampuan perangkat selular, format grafik berbasis vektor menjadi pilihan mengingat skalabilitas dan portabilitasnya. Grafik 2D berbasis vektor dapat digunakan untuk laporan berita lalu lintas dan cuaca, pemetaan, navigasi, *multimedia messaging*, animasi dan grafik interaktif, dan *content* hiburan. Sebagai *open-standard* baru di bidang grafik vektor, *Scalable Vector Graphics* (*SVG*) telah digunakan secara luas untuk representasi data grafik 2D dalam berbagai bidang. Beberapa penelitian memanfaatkan format *SVG* ini untuk visualisasi data ilmiah, diantaranya visualisasi sensus data *online*, data medis, aplikasi *Geographical Information System* (*GIS*), navigasi dan *distance-education/learning*. *SVG* dianggap dapat menyediakan grafik dan interaktifitas dokumen yang lebih baik, serta memungkinkan untuk pemisahan *layer-layer content* [3].

Masalah yang umumnya menjadi hambatan dalam pengembangan aplikasi pada lingkungan *mobile* adalah *issue-issue* yang terkait dengan keterbatasan perangkat *mobile* yang masih perlu diperhatikan terutama untuk pengembangan aplikasi di bidang pemrosesan data grafik yang kompleks, antara lain ukuran layar yang kecil, ketersediaan *power* dan kemampuan komputasional yang terbatas, serta keterbatasan metode *user-input*. Selain itu, komunikasi *wireless* sendiri juga mempunyai masalah dalam hal *bandwidth* yang kecil, kestabilan koneksi, serta gangguan sinyal [4]. Untuk aplikasi atau layanan yang bersifat *online*, masalah biasanya belum muncul pada proses transmisi file grafik vektor dua dimensi yang berukuran kecil, tetapi jika ukuran filenya besar, misalnya transmisi file sebesar 2MB melalui koneksi *GPRS*, proses transmisi akan berlangsung lama sehingga berpotensi terganggu oleh adanya *service-interruption* atau terjadinya *error* pada saat *download* [5]. Pada aplikasi-aplikasi grafik yang mempunyai fungsionalitas *offline*, biasanya masalah terbesar adalah pada proses *parsing* dan *rendering* yang relatif memberatkan memori perangkat *mobile*.

Perancangan aplikasi layanan *content* pada penelitian ini akan menggunakan sistem representasi grafik multi-level berbasis *SVG* yang diterapkan pada suatu bentuk format *content* majalah *mobile*. Cara ini memungkinkan suatu *content* utuh yang idealnya berukuran besar dapat dipisah-pisahkan ke dalam beberapa *sub-content* berukuran kecil dengan tetap mempertahankan hubungan antar tiap *sub-content* tersebut. Dengan ukuran yang lebih ringan, diharapkan aplikasi *mobile* dapat memperoleh performa yang baik terutama pada proses *parsing* dan *rendering content*.

2. Kajian Pustaka

2.1 Visualisasi Grafik

Beberapa penelitian telah dilakukan berkaitan dengan pengelolaan gambar 2D berukuran relatif besar pada lingkungan komputasi *mobile*, baik itu pada data raster maupun grafik vektor. Pada penelitian yang telah dilakukan (Karstens, Rosenbaum dan Schumann, 2004) didapatkan pernyataan-pernyataan diantaranya bahwa *panning* lebih cepat daripada *scaling*, *scaling* sederhana pada data raster relatif cepat tetapi menyebabkan kualitas presentasi menjadi rendah, semakin bertambahnya manipulasi pada data berarti semakin bertambahnya waktu pemrosesan data, penanganan data *raster* lebih cepat dari data vektor.

Menurut pernyataan-pernyataan di atas, sebenarnya penggunaan data *raster* untuk beberapa kondisi akan lebih masuk akal karena data raster dapat di-*load*, diproses, kemudian ditampilkan dengan sangat cepat, tetapi jika gambar yang ingin ditampilkan mengandung detail yang lebih banyak dan memerlukan kualitas yang lebih bagus, maka format data vektor adalah pilihan yang lebih baik, terutama untuk penggunaan dalam lingkungan *mobile*, dimana pengelolaan file grafik berukuran besar secara layak masih sulit direalisasikan.

Faktor keterbatasan layar pada perangkat *mobile* menyebabkan suatu bentuk informasi yang dapat ditampilkan dalam satu kali kesempatan mungkin hanya sebagian kecil dari keseluruhan informasi yang sesungguhnya. Melihat kenyataan ini, informasi utuh yang berukuran relatif besar sebenarnya dapat dipisah-pisahkan

agar dapat lebih efisien. Dalam hal ini diperlukan metode tertentu untuk menghubungkan bagian-bagian informasi ini untuk keperluan navigasi dan orientasi. Salah satu metode yang dikenal adalah dengan memvisualisasikan relasi antarbagian yang ada. Struktur seperti ini dapat ditampilkan sebagai grafik, dimana cabang-cabang grafik ini akan mempresentasikan beberapa bagian informasi. Namun, akan tetap sulit untuk menjaga orientasi jika belum pasti apakah keseluruhan cabang grafik dapat ditampilkan. Di lain pihak, metode *hierarchical structures* (struktur hirarki) memungkinkan kontrol yang lebih efisien dari cabang-cabang informasi yang ingin ditampilkan. Contoh struktur yang disusun sebagai hirarki ini dapat dilihat pada struktur *file system*. Dengan struktur yang sejenis, metode *hierarchic representation* dapat diterapkan pada perangkat *mobile* untuk keperluan representasi grafik. Jika sebuah grafik dapat disusun sebagai struktur hirarki, ini akan memberikan keuntungan pada sistem representasi grafik tersebut.

2.2 Scalable Vector Graphics (SVG)

SVG (Scalable Vector Graphics) merupakan bahasa sekaligus format file untuk menampilkan grafik dua dimensi berbasis vektor. *SVG* dikembangkan berdasarkan bahasa *XML (eXtensible Markup Language)*, sehingga perangkat-perangkat lunak yang dapat menginterpretasi *XML* akan dapat pula menginterpretasi *SVG*.

SVG dapat digunakan untuk membuat tiga jenis objek grafik, yaitu: path (terdiri dari garis lurus dan kurva), gambar, dan teks. *SVG* dapat mengkreasikan sebuah grafik yang terdiri dari banyak vektor yang berbeda-beda. Kelebihan *SVG* yang berbasis vektor ini terutama adalah gambar tidak akan kehilangan kualitasnya apabila diperbesar atau diperkecil (*scalable*), karena dibuat berdasarkan vektor (*vector*), bukan *pixel* (seperti format grafik pada umumnya, *GIF*, *JPEG* dan *PNG*). *SVG* juga dapat menampilkan efek bayangan, gradasi warna atau juga pencahayaan. Selain itu, animasi juga dapat dikembangkan *SVG*, sesuatu yang tidak dimiliki oleh *GIF*, *JPEG* dan *PNG*. Hal ini dimungkinkan dengan integrasi *DOM (Document Object Model)*. Jadi, grafik *SVG* dapat dianimasikan melalui perintah *script*.

a. Mobile SVG

SVG 1.0 merupakan versi pertama yang dikeluarkan oleh *World Wide Web Consortium (W3C)* dilanjutkan dengan *SVG 1.1* yang merupakan modularisasi dari *SVG 1.0*. Melihat maraknya tuntutan pasar industri *mobile* serta kuatnya dukungan dan permintaan yang berasal dari pengembang maupun komunitas-komunitas *SVG*, *W3C* menyadari perlunya membuat format *SVG* yang sesuai untuk menampilkan grafik pada perangkat *mobile*.

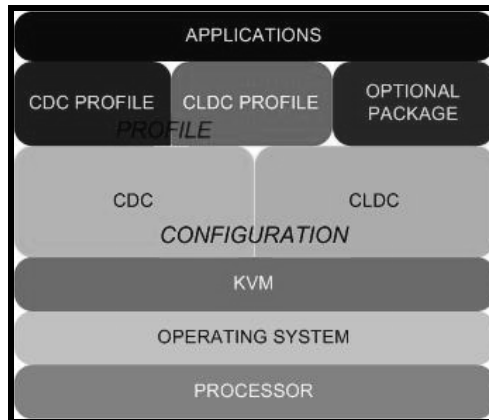
Pada Desember 2003 *W3C* melalui *SVG Working Group* merilis dua profile *Mobile SVG* yaitu *SVG Tiny (SVGT)* yang ditujukan untuk perangkat *mobile* dengan sumberdaya terbatas seperti telepon selular dan *SVG Basic (SVGB)* yang ditujukan untuk level perangkat *mobile* yang lebih tinggi seperti *PDA*. *Content* dari *SVGB* dan *SVGT* dapat berupa dokumen *SVG* yang berdiri sendiri (*stand-alone*) atau *document-fragment* yang ditempelkan diantara dokumen *XML*.

2.3 Really Simple Syndication (RSS)

RSS adalah sejenis format data berbasis XML yang digunakan untuk mempublikasikan digital content yang sering di-update, seperti blog, berita, dan podcast. Informasi yang disediakan oleh suatu website ini berbentuk RSS yang berupa file XML dan biasa disebut RSS Feed.

2.4 Java ME (Micro Edition)

Java ME merupakan *platform* pemrograman Java untuk perangkat *mobile*. Arsitektur Java ME secara umum dapat dilihat pada Gambar 2.1.



Gambar 2.1. Arsitektur Platform Perangkat Lunak Java ME

Optional Package merupakan API (*Application Programming Interface*), tetapi berbeda dengan *profile*, *Optional Package* tidak berisi paket lengkap untuk membangun *application environment*. *Optional Package* selalu digunakan bersama dengan *configuration* atau *profile*. *Optional Package* menambahkan fungsi-fungsi perangkat tertentu yang tidak cukup universal untuk dijadikan bagian dari standar *profile*, atau justru karena fungsi tersebut tidak hanya dapat digunakan untuk satu standar *profile*. Untuk dapat menggunakan aplikasi yang memakai *Optional Package* pada suatu perangkat, *Optional Package* yang bersangkutan harus telah diintegrasikan dalam sistem. Biasanya yang menentukan *Optional Package* manakah yang tersedia pada suatu perangkat adalah vendor-vendor pembuat perangkat keras.

2.4.1 Optional Package JSR 226

Pada JSR 226 dijelaskan spesifikasi final API untuk SVG (*Scalable 2D Vector Graphics*). API ini ditujukan untuk perangkat mobile yang memiliki sumber daya terbatas dalam hal memori, ukuran layar dan kemampuan komputasionalnya. Tujuan dari spesifikasi ini adalah untuk menentukan *package* API tambahan untuk rendering format gambar dua dimensi berbasis vektor, terutama terfokus pada format SVG Tiny.

JSR 226 terdiri dari kelas-kelas untuk penciptaan dan rendering image vektor dan kelas-kelas untuk memanipulasi komponen XML dari suatu image vektor sebagai bagian dari pohon hirarki DOM. Kelas-kelas ini didefinisikan dalam package `javax.microedition.m2g` dan `org.w3c.dom.svg`.

2.4.2 JSR 172

Data dengan format XML yang dikirimkan ke suatu *mobile client* harus diterjemahkan terlebih dahulu dengan membuat coding yang akan bertindak sebagai *Parser* dokumen XML tersebut. Untuk menghindari setiap pengembang program harus membuat kode khusus untuk keperluan *parsing XML*, dibuat standar *optional package* untuk dukungan XML Parsing pada platform Java ME yaitu JSR 172. JSR 172 memiliki dua tujuan utama, yaitu menentukan standar *optional package* yang mendukung XML Parsing dan standar *optional package* untuk memberikan dukungan akses *web service* berbasis XML untuk CDC maupun CLDC.

2.4.2.1 JAXP Subset (Java API for XML Processing)

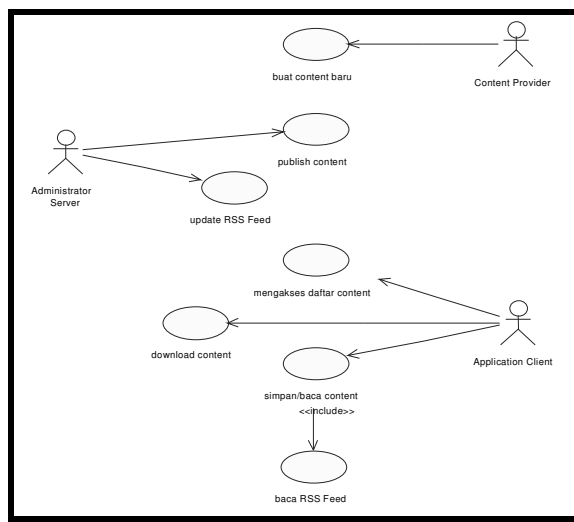
JAXP XML Parsing API berbasis pada aturan SAX 2.0 (*Simple API for XML version 2.0*). JAXP Subset berbasis pada aturan-aturan SAX 2.0, mendukung XML namespaces, mendukung character encoding UTF-8 dan UTF-16, mendukung DTD (*Document Type Definitions*), namun tidak mendukung DOM (*Document Object Model*) dan XSLT (*Extensible Stylesheet Language Transformations*) karena dianggap terlalu berat untuk perangkat *mobile*.

3. Perancangan dan Implementasi

Pada dasarnya perancangan sistem dapat dibagi menjadi dua bagian yaitu *client* dan server. Representasi grafik multi-level terjadi pada sisi *client*, sedangkan server bertindak sebagai penyedia *content* untuk implementasi sistem secara utuh.

3.1 Gambaran Sistem Secara Keseluruhan

Diagram *use-case* sistem secara keseluruhan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram *Use-case* Sistem

Melihat gambar 3.1 di atas, *content* merupakan olahan perancangan majalah yang dihasilkan oleh *Content Provider*, yaitu penyedia layanan informasi majalah yang merangkum isi majalahnya dalam format *mobile* berbasis SVG, menurut *template* tertentu. *Server* bertugas mengelola persediaan *content* yang didapat dari *Content Provider* dan menerbitkan *update menu* setiap ada *content* baru, serta meletakkan meletakkan berkas *content* pada *URL*. Aplikasi *client* dapat mengambil informasi menu *downloadable content* kemudian memilih *content* yang ingin di-download. *Content* yang telah di-download dapat disimpan untuk dibaca secara *offline*.

3.2 Rancangan Sistem Representasi Grafik Multi-Level berbasis SVG pada Content Majalah Mobile

Sistem representasi grafik multi-level berbasis SVG yang diterapkan pada sebuah *content* majalah *mobile* memungkinkan suatu *content* utuh yang idealnya berukuran besar dapat dipisah-pisahkan ke dalam beberapa *sub-content* berukuran kecil dengan tetap mempertahankan hubungan antar tiap *sub-content* tersebut. Pemisahan ini dapat dilakukan dengan aturan yang ditetapkan menurut kebutuhan. Misalnya pemisahan menurut jenis elemen grafik dan pemisahan menurut isi *content* yang dapat direalisasikan dengan adanya elemen *group* dalam grafik SVG. Pada *content* majalah *mobile* ini digunakan metode yang kedua, dengan alasan agar *script* dari setiap *sub-content* tetap dapat ditampilkan sebagai gambar sesuai keinginan pembaca dengan meminimalisir proses pembacaan file pada *device*.

Contoh format *content* majalah dan pemisahannya adalah sebagai berikut:

```
<svg>
  <g display="none" id="descriptions">
  <g id="title"><text>...</text></g>
  <g id="parent"><text>self</text></g>
  ...
</g>
<g display="none" id="list of content">
<text>...</text>
...
</g>
<g id="page_1">
<g id="judul_page_1">...</g>
<g id="inset_page_1" transform="translate(4,-1)"> ...</g>
...
</g>
<g id="page_2">...</g>
...
</svg>
```

Gambar 3.2 Contoh *Content* Majalah Utuh

Content utuh diatas di-*parsing* dan diambil informasi utamanya untuk dijadikan *root content* seperti terlihat di bawah ini:

```
<svg>
  <g display="none" id="descriptions">
    <g id="title"><text>...</text></g>
    <g id="parent"><text>self</text></g>
    ...
  </g>
  <g display="none" id="list of content">
    <text>...</text>
    ...
  </g>
  <g display="none" id="graphics">
    <text>page_1.svg</text>
    <text>page_2.svg</text>
    ...
  </g>
</svg>
```

Gambar 3.3 Contoh *Root Content*

Parsing selanjutnya tidak terbatas sampai berapa level *sub-content* yang ingin diaplikasikan. Sistem akan membaca apakah pada *sub-content* terdapat elemen *group* (`<g></g>`) dengan atribut `id="graphics"`. Keberadaan elemen menandakan bahwa content tersebut bukan *sub-content* level terakhir, melainkan sistem akan mencari *sub-content* level berikutnya menurut *childnodes* dari elemen tersebut. *Sub-content* yang tidak memiliki elemen *group* dengan `id="graphics"` menandakan *sub-content* level terakhir dan dapat ditampilkan pada layar.

```
<svg>
  <g display="none" id="descriptions">
    <g id="title"><text>page_1</text></g>
    <g id="parent"><text>root</text></g>
  </g>
  <g id="graphics" display="none">
    <text>page_1_judul.svg</text>
    <text>page_1_headline.svg</text>
    <text>page_1_inset.svg</text>
    <text>page_1_isi.svg</text>
  </g>
</svg>
```

Gambar 3.4 Contoh *Sub-Content* page_1.svg

```
<svg>
  <g display="none" id="descriptions">
    <g id="title"><text>inset</text></g>
    <g id="parent"><text>page_1</text></g>
  </g>
  <g id="inset" transform="translate(4,-1)">
    <rect x="17.99" y="-141.99" width="73.99" height="7.99" fill="#cf3"
      stroke="#000" stroke-linejoin="bevel" stroke-miterlimit="4"/>
    ...
  </g>
</svg>
```

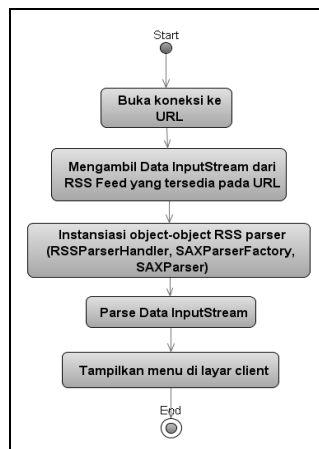
Gambar 3.5 Contoh *Sub-Content* page_1_inset.svg (level terakhir)

3.3 Perancangan Perangkat Lunak Client

Perangkat lunak pada sisi client terdiri atas Modul *Download*, *RSSFeedParser*, *ContentBrowser* dan Modul *ContentMerging & Rendering*.

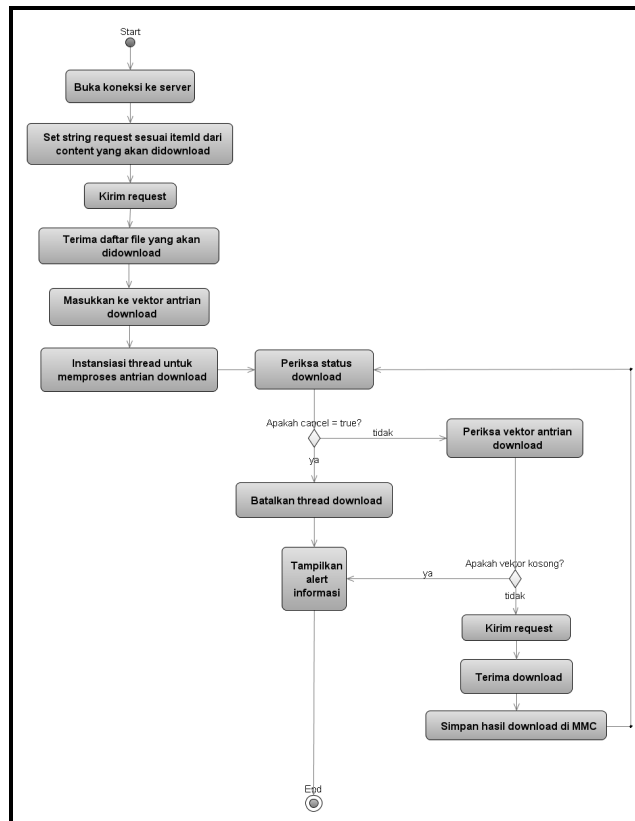
i. Modul *Download* dan Modul *RSSFeedParser*

Modul *Download* digunakan untuk mengambil informasi menu *downloadable content* dan melakukan proses download sesuai *request* dari *client*. Informasi menu diambil melalui *RSSFeedParser* yang merupakan implementasi JSR172, sedangkan proses *download* melalui sebuah *thread* yang menggunakan koneksi *HttpConnection*.



Gambar 3.6 Diagram Aktivitas Pengambilan Informasi Menu *Downloadable Content*

Proses *thread download* digambarkan pada diagram aktivitas berikut:



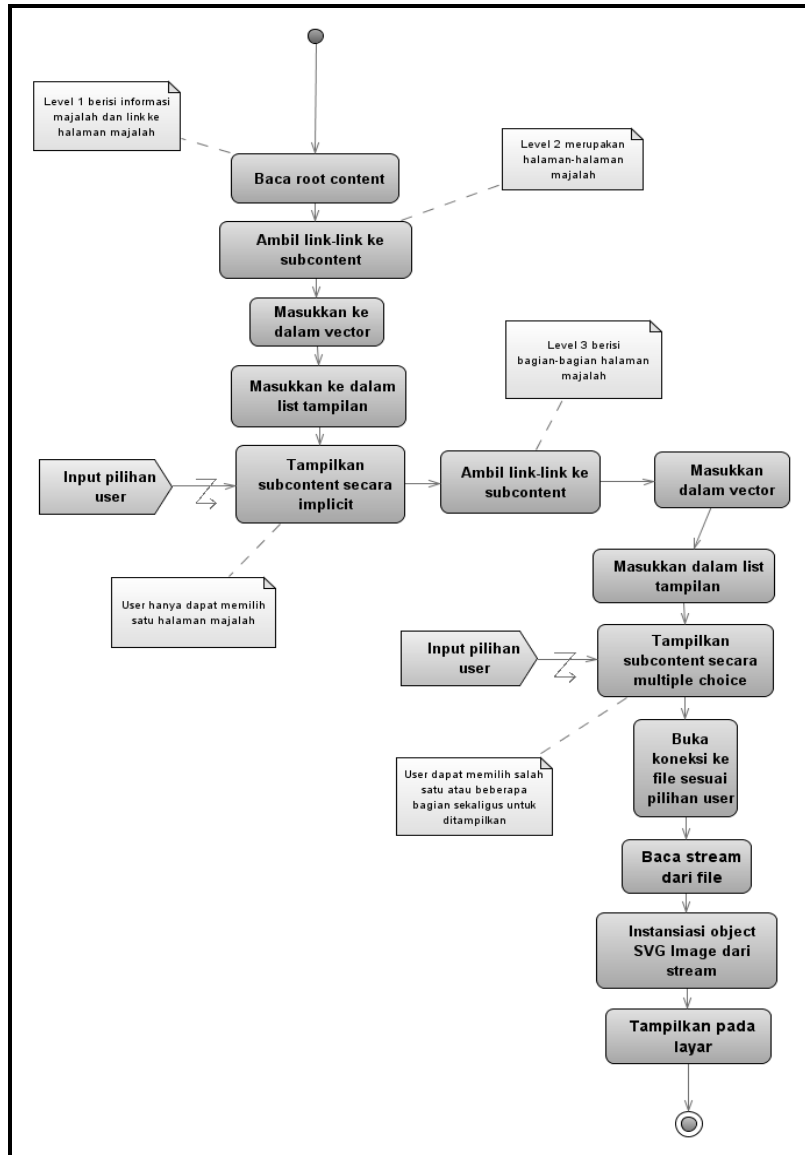
Gambar 3.7 Diagram Aktivitas Modul *Download* di sisi *Client*

ii. Modul *ContentBrowser*

Modul *ContentBrowser* berfungsi untuk mengakses *file system* dari *device* dan mencari file berekstensi *.mag* yang merupakan file utama dari *content* majalah. Modul ini diimplementasikan sebagai thread dan menggunakan *optional package JSR 75* yaitu *FileConnection API*.

iii. Modul *ContentMerging & Rendering*

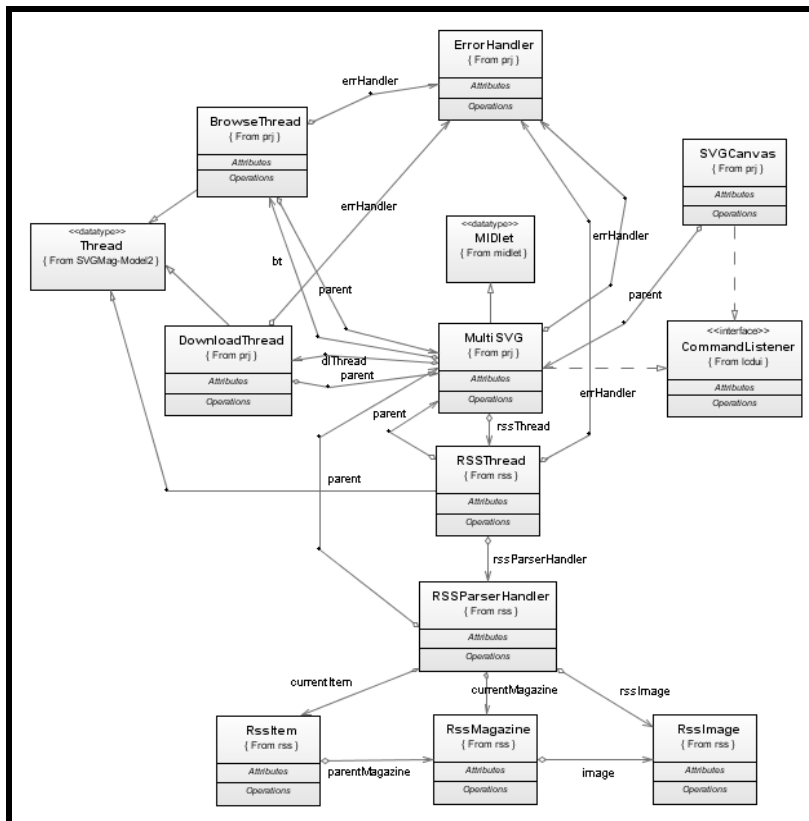
Modul *ContentMerging & Rendering* berfungsi untuk menerjemahkan struktur grafik *content*, mencari *link* antara *subcontent* dan menampilkannya pada layar. Pada Gambar 3.8 ditunjukkan gambaran proses yang terjadi pada Modul *ContentMerging & Rendering* :



Gambar 3.8 Aktivitas Diagram Modul *ContentMerging & Rendering* di sisi *Client*

iv. Diagram Kelas

Dalam diagram kelas ini, diperlihatkan hubungan antar kelas dan penjelasan singkat tiap-tiap kelas. Diagram kelas untuk sistem pada sisi *client* dapat dilihat pada Gambar 3.9.



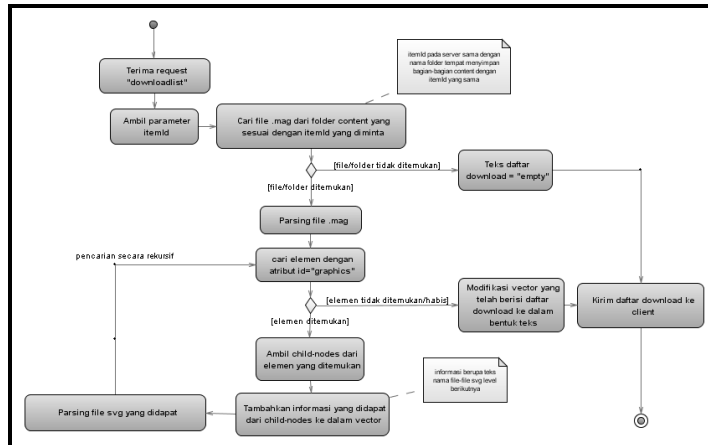
Gambar 3.9 Diagram Kelas Sistem *Client-side*

3.4 Perancangan Perangkat Lunak Server

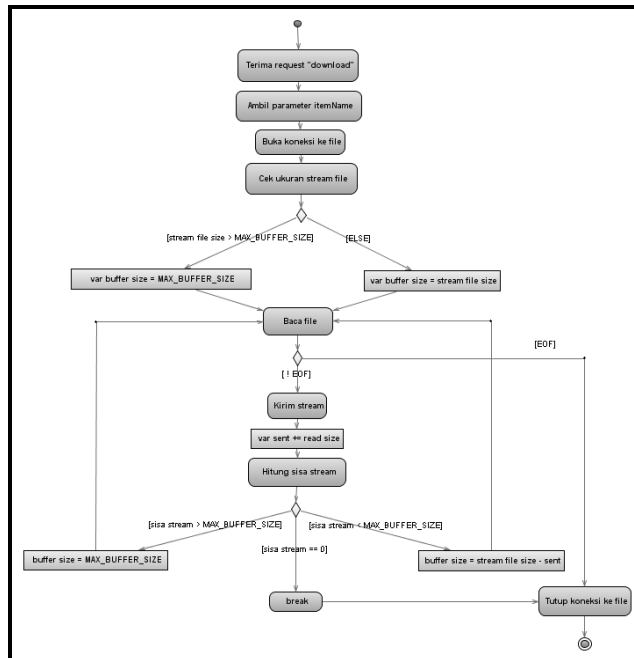
Perangkat lunak pada sisi *server* terdiri atas Modul *DownloadProcessor* dan Modul *ContentParsing*. Modul utamanya adalah *DownloadProcessor*.

3.4.1 Modul *DownloadProcessor*

Modul *DownloadProcessor* pada *server* terdiri dari submodul untuk memproses permintaan *download* dan submodul untuk mengirim *content*.



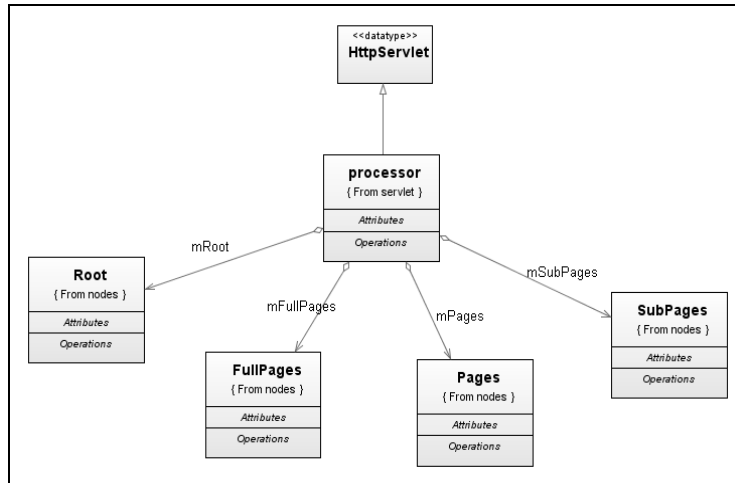
Gambar 3.10 Modul *DownloadProcessor* pada Saat Memproses *Request DownloadList*



Gambar 3.11 Modul *DownloadProcessor* pada Saat Memproses *Request Download*

v. Diagram Kelas

Diagram kelas untuk sistem di sisi *server* dapat dilihat pada Gambar 3.12.



Gambar 3.12 Diagram Kelas Sistem *Server-side*

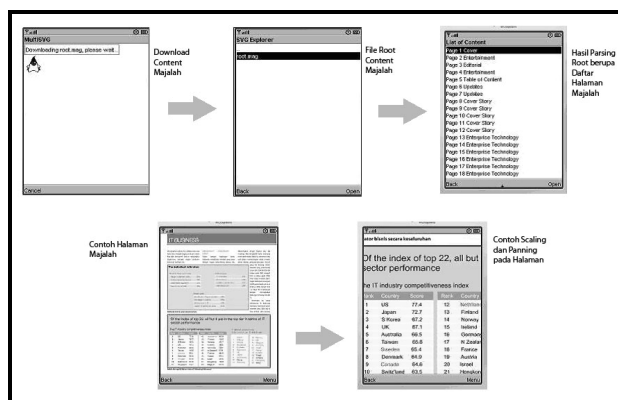
4. Hasil Pengujian dan Pembahasan

Sistem pada sisi *client* dibangun dengan platform *Java Micro Edition* (MIDP 2.1, CLDC 1.1) dan menggunakan *optional package* untuk *Scalable Vector Graphics* (JSR 226), *Web Service/XML Parsing API* (JSR 172) dan *PDA Optional API* (JSR 75). Pada sisi *server*, sistem dibangun dengan platform *Java Enterprise Edition* (JEE) dan menggunakan web container *Glashfish V2*. Untuk percobaan digunakan *Sun Wireless Toolkit 2.5*.

4.1 Pengujian Sistem

4.1.1 Fitur Aplikasi

Berdasarkan rancangan yang telah dipaparkan, dikembangkan sebuah aplikasi majalah pada sarana bergerak (*mobile magazine*) yang fungsionalitasnya secara umum dapat dilihat pada Gambar 4.1.



Gambar 4.1. Aplikasi Mobile Magazine dengan Representasi Grafik Multi-Level Berbasis SVG

Tabel 4.1 menyajikan hasil pengamatan terhadap performa *rendering content* multi-level majalah mini berformat SVG pada aplikasi yang dibuat.

No.	Ukuran Halaman / File (KB)	Jumlah vector primitive (object)	Decoding (ms)	Event (ms)			
				Initial Opening	Zoom In	Zoom Out	Panning
1.	33	91	743	196	198	190	148
2.	62	1	849	136	134	127	95
3.	65	1	843	125	174	109	84
4.	91	1	1072	116	131	98	126
5.	99	315	1602	818	778	768	644
6.	106	567	1802	1461	1306	1355	1122
7.	107	648	2269	1781	1946	1647	1341
8.	123	329	1412	865	857	839	722
9.	127	157	2271	730	725	694	598
10.	128	786	2274	1866	1768	1744	1580
11.	138	381	1803	793	1298	1436	1098

Tabel 4.1. Performa *Rendering Content* Multi-Level

Dari hasil tabel diatas dapat diketahui bahwa waktu yang dibutuhkan untuk proses *scaling*, *panning* dan proses menampilkan *image* ke layar dipengaruhi oleh jumlah *vector primitive* yang terkandung dalam *content*. Sedangkan ukuran file turut mempengaruhi proses *decoding data content* menjadi obyek *image*.

4.1.2 Verifikasi Aplikasi menurut Standar *Unified Testing Criteria for Java(TM) Technology-based Applications for Mobile Devices Version 2.1* dari *Sun Microsystems*

Pada pengujian ini akan dilakukan serangkaian tes terhadap berbagai kasus yang dicobakan pada aplikasi, yang dikelompokkan ke dalam 10 kategori yang berbeda, yaitu *Application Characteristics*, *Stability*, *Application Launch*, *User Interface Requirements*, *Localization*, *Functionality*, *Connectivity*, *Personal Information Management*, *Security*, *Retesting*. Keberhasilan/kegagalan aplikasi ditentukan oleh kemampuan aplikasi untuk melewati semua kategori testing. Setiap tes memiliki rating yang sama, sehingga tidak diperlukan scoring.

Berdasarkan pengujian yang dilakukan, dihitung jumlah poin-poin pengujian yang berhasil dan yang tidak berhasil sehingga didapatkan tingkat keberhasilan pengujian $31/32 \times 100\% = 96,8\%$

5. Kesimpulan dan Saran

Sistem melewati pengujian verifikasi aplikasi menurut standar *Unified Testing Criteria for Java(TM) Technology-based Applications for Mobile Devices Version 2.1* dari **Sun Microsystems** dengan tingkat keberhasilan sebesar 93,75%. Kekurangan pada poin pengujian *Application Stability* berkaitan besarnya memori yang dibutuhkan oleh aplikasi. Kecepatan pemrosesan content dipengaruhi oleh faktor-faktor ukuran file, jumlah vektor primitive dan kompleksitas content.

Untuk menindaklanjuti hasil perancangan sistem dalam penelitian ini, beberapa hal perlu dilakukan, mengingat masih terdapatnya beberapa keterbatasan dalam penelitian ini, antara lain, *library Java ME* yang masih belum dapat membaca *system font* dari sistem operasi perangkat selular tertentu, sehingga data *font* harus ditambahkan sebagai elemen *embedded glyphs*. Hal ini mengakibatkan ukuran file bertambah secara signifikan.

Beberapa saran pengembangan yang dapat dilakukan diantaranya penggunaan *platform* pengembangan perangkat lunak selain *Java*, seperti *Symbian* dan pengembangan *web service* untuk sisi server, penggunaan server *bluetooth* sebagai pengganti koneksi GPRS, maupun penambahan proteksi pada *content* untuk keperluan royalti. Contoh *content* yang digunakan pada penelitian ini menggunakan *content SDA Asia Magazine* versi digital yang berformat PDF dan dikonversi ke dalam format *SVG* menggunakan *Adobe Illustrator CS3*. Konversi ini mengakibatkan desain *content* dan pengaturan teks kurang terkontrol. Jika *content* dapat dibuat dari desain awal oleh *content provider* misalnya, kualitas *content* dapat lebih baik dan ukuran file dapat lebih dikontrol.

Daftar Pustaka

- [1] Timur, Anton. Mencari Konten 3G Unggulan. <http://sudrajat.livejournal.com/600.html> . Diakses tanggal: 25 Maret 2008.
- [2] Antara News. Menkominfo : “Kontribusi Industri Konten Indonesia Minim”. <http://www.antaranews.co.id>. Diakses tanggal: 25 Maret 2008.
- [3] Lee, Sangmi, Fox, Ko, Wang dan Qiu, 2002, “Ubiquitous Access for Collaborative Information System using SVG”, Proc. of SVG Open/Carto.net Developers Conference, 2002
- [4] Rosenbaum, R., Schumann, dan Tominski, Ch., 2004, “Presenting Large Graphical Contents on Mobile Devices-Performance Issues”, P.C. Deans: E-Commerce and M-Commerce Technologies, IRM Press, 2004.
- [5] Xiaoyong Su, Prabhu B. S., Chu Chi-Cheng, Gadh Rajit, “Scalable Vector Graphics (SVG) Based Multi-Level Graphics Representation For Engineering Rich-Content Exchange In Mobile Collaboration Computing Environments”, Journal of Computing and Information Science in Engineering, 2006.
- [6] Karstens, B., Rosenbaum, dan Schumann, 2004, “Presenting Large and Complex Information Setson Mobile Handhelds”, Idea Group Inc., 2004.
- [7] Andersson, Ola, dkk, “Mobile SVG Profiles : SVG Tiny and SVG Basic”, W3C Recommendation, <http://www.w3.org/TR/SVGMobile>, 2003.
- [8] Downes, Barry, dkk, “Mobile Operator Publishing and Entertainment Platform”, Proceedings of the International Conference on Mobile Business (ICMB’05), 2005.
- [9] Giguere, Eric, “J2ME Optional Packages”, 2002.
- [10] Ortiz, Enrique, “Introduction to J2ME Web Services”, <http://developers.sun.com/mobility/apis/articles/wsa/>, 2004.
- [11] Chonoles, Michael Jesse, Schardt, “UML 2 for Dummies”, Hungry Minds, 2003.
- [12] Nugroho, Adi, “Analisis dan Perancangan Sistem Informasi Menggunakan Metodologi Berorientasi Objek”, Penerbit INFORMATIKA, Bandung, 2004.

Pemodelan Sistem Informasi pada CV Cihanjuang Inti Teknik dengan Menggunakan Zachman Framework

Meliana Christianti, Felly Dias Try

Jurusan Teknik Informatika

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164

Email: hallomell@yahoo.com, fly4ulove@gmail.com

Abstract

The development of information technology has been increased time to time, so that it has a big influence in many aspects of business development. In this case, the organizer company needs a form of documentation to show business development from time to time. There are some methodologies to document a data, one of them is Zachman framework methodology.

Zachman framework is one of enterprise architectures that is used to describe the process in organizations. The Framework has six columns: what column, how column, where column, who column, when column, and why column. What column describes about business entities and data relation in the organization. How column defines the business processes of the organization. Where column describes the location of business. Who column describes the human resources of the organization. When column describes the major event of the company. Why column defines vision, mission, strategic, target, and long term planning of the organization. The documentation had been done from October 2007 until January 2008.

Keyword: Zachman framework, enterprise architecture, documentation, organization.

1. Latar belakang

Dilihat dari perkembangan saat ini, teknologi dan sistem informasi bukan suatu hal yang baru tetapi terus berkembang pesat. Hal ini sangat berpengaruh bagi organisasi besar, organisasi menengah maupun organisasi kecil. Sistem informasi dan teknologi merupakan hal yang paling dasar dalam mengembangkan usaha ke arah yang lebih baik. Sehingga dapat meningkatkan daya saing diantara para pebisnis.

Dalam pengembangan bisnis perlu adanya pendokumentasian yang terorganisir dengan baik. Agar perkembangan perusahaan dapat dilihat perkembangannya dari waktu ke waktu, maka penulis mencoba untuk melakukan pendokumentasian pada CV Cihanjuang Inti Teknik.

1.1. Perumusan Masalah

Dalam pemanfaatan teknologi seperti teknologi informasi sebaiknya dilakukan dengan perencanaan. Namun, banyak perusahaan – perusahaan yang sedang berkembang tidak melakukan hal tersebut. Hal ini disebabkan tidak adanya pendokumentasian yang baik. Sehingga sering terjadi perubahan – perubahan kebijakan bisnis yang harus dilakukan dari awal, yang sebenarnya dapat diatasi dengan perbaikan – perbaikan pada bagian – bagian tertentu.

Dalam menganalisis sistem informasi dan teknologi, perumusan masalah dalam penulisan laporan ini adalah sebagai berikut :

1. Apakah manfaat dari pendokumentasian *enterprise architecture planning* pada CV Cihanjuang Inti Teknik ?

2. Bagaimana Zachman Framework dapat digunakan untuk memberikan deskripsi kerja dan struktur organisasi di CV Cihanjuang Inti Teknik?
3. Bagaimana Zachman Framework dapat digunakan di CV Cihanjuang Inti Teknik untuk mendefinisikan proses bisnis dengan jelas?

1.2. Tujuan

Framework yang digunakan dalam melakukan analisis dan dokumentasi *Enterprise Architecture* adalah *Zachman framework*. Pendokumentasian dengan menggunakan metodologi *Zachman framework* dapat mendefinisikan organisasi secara lengkap. Hasil akhir yang diharapkan yaitu pendokumentasian *Enterprise Architecture* dapat membantu perusahaan dalam mengambil kebijakan – kebijakan untuk pengembangan perusahaan, menggambarkan kondisi perusahaan saat ini dan perencanaan yang akan datang, dapat membantu dalam mengontrol seluruh kegiatan bisnis yang dilaksanakan oleh perusahaan, dan dapat mengendalikan sumber daya perusahaan yang ada, serta dapat membantu perusahaan dalam menentukan rencana yang akan dilakukan dalam mengembangkan bidang usaha.

1.3. Pembatasan Masalah

Dalam penulisan laporan ini memiliki batasan masalah antara lain sebagai berikut:

1. Pemodelan sistem informasi menggunakan metodologi *Zachman framework*, karena *Zachman framework* menggambarkan arsitektur *enterprise* lebih lengkap.
2. Analisis dilakukan pada proses – proses yang berkaitan dengan divisi teknik CV Cihanjuang Inti Teknik.
3. Periode analisis kegiatan yang didokumentasikan adalah pada bulan Oktober 2007 sampai dengan Januari 2008.
4. Berikut ini pembatasan masalah dalam pendokumentasian *Zachman framework*.
 - a. Kolom *What* membahas mengenai data yang ada pada CV Cihanjuang Inti Teknik. Bagian yang diuraikan pada kolom ini adalah *scope*, *enterprise model*, dan *system model*.
 - b. Kolom *How* membahas mengenai proses – proses bisnis yang terjadi di CV Cihanjuang Inti Teknik. Bagian yang diuraikan pada kolom ini adalah *scope*, *enterprise model*, dan *system model*.
 - c. Kolom *Where* membahas mengenai lokasi bisnis CV Cihanjuang Inti Teknik. Bagian yang diuraikan pada kolom ini adalah *scope*, *enterprise model*, *system model*, dan *technology model*.
 - d. Kolom *Who* membahas mengenai sumber daya manusia yang berperan pada CV Cihanjuang Inti Teknik. Bagian yang diuraikan pada kolom ini adalah *scope*, *enterprise model*, *system model*, *technology model*, dan *functioning system*.
 - e. Kolom *When* membahas mengenai waktu dan kegiatan yang dilakukan CV Cihanjuang Inti Teknik. Bagian yang diuraikan pada kolom ini adalah *scope*, *enterprise model*, *system model*, dan *functioning system*.
 - f. Kolom *Why* membahas mengenai hal – hal yang ingin dicapai oleh CV Cihanjuang Inti Teknik. Bagian yang diuraikan pada kolom ini adalah *scope*, dan *enterprise model*.

Demikian batasan masalah yang disampaikan penulis untuk digunakan dalam pengerjaan kerja praktek.

2. Pengenalan Zachman framework

Zachman framework merupakan salah satu *Enterprise Architecture* yang digunakan untuk memodelkan bisnis yang ada dalam suatu perusahaan. Orang yang mengemukakan ide ini adalah John Zachman pada tahun 1987, dengan nama *Zachman Framework for Enterprise Architecture and Information Systems Architecture*^[5].

Zachman Framework mempertimbangkan analogi bahwa proses pengembangan system informasi sama seperti membangun sebuah rumah. Kita harus menentukan rumah seperti apa yang ingin ditempati, apa yang disukai dari rumah tersebut, berapa banyak kamar yang akan di inginkan, dimana lokasinya, dan pertimbangan lainnya dan juga mendefinisikan secara jelas dan berbeda mengenai tiga macam arsitektur, yaitu data, proses (aplikasi), dan jaringan (teknologi).

Pendokumentasian pada *Zachman Framework* lebih lengkap, yaitu terdapat enam baris dan enam kolom yang mencakup, *What* (aspek data), *How* (aspek fungsi), *Where* (aspek jaringan), *Who* (aspek sumber daya manusia), *When* (aspek waktu), *Why* (aspek motivasi), dan. Kemudian pada bagian baris yaitu: *scope* (sudut pandang *planner*), *enterprise model* (sudut pandang *owner*), *system model* (sudut pandang *designer*), *technology model* (sudut pandang *builder*), *components* (sudut pandang *subcontractor*), dan *functioning system* (sudut pandang *enterprise*).

3. Analisis dan Perancangan

3.1. Kolom WHAT

Bagian yang diuraikan pada kolom *What* yaitu *scope*, *enterprise model*, dan *system model*. Berikut ini penjelasan *scope*, *enterprise model*, dan *system model*.

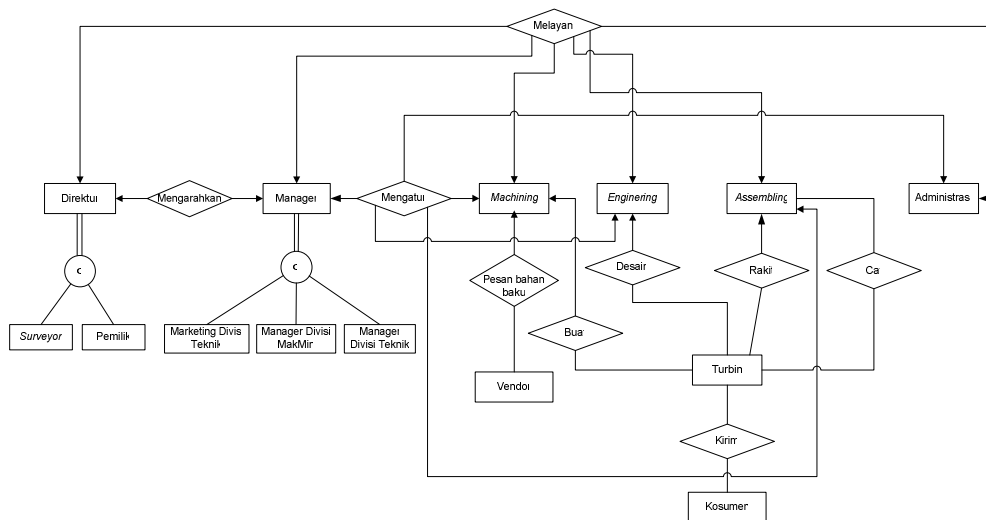
3.1.1. SCOPE : PLANNER PERSPECTIVE

Pada bagian *scope* menguraikan mengenai entitas bisnis yang penting yang ada di CV Cihanjuang Inti Teknik. Berikut ini entitas yang terlibat pada CV Cihanjuang Inti Teknik :

- | | | |
|----------------------|---------------------------|----------------------|
| 1. Pemilik | 6. <i>Assembling</i> | 11. Karyawan |
| 2. Direktur | 7. <i>Quality control</i> | 12. Turbin |
| 3. <i>Surveyor</i> | 8. <i>Marketing</i> | 13. Konsumen |
| 4. <i>Manager</i> | 9. <i>Vendor</i> | 14. <i>Machining</i> |
| 5. <i>Enginering</i> | 10. Administrasi | |

3.1.2. ENTERPRISE MODEL : OWNER PERSPECTIVE

Pada bagian *enterprise model* menggambarkan hubungan antara entitas bisnis yang terlibat pada CV Cihanjuang Inti Teknik saat ini. Berikut ini Diagram relasional entitas bisnis tersebut.



Gambar III.1 Diagram relasional entitas bisnis

Berikut ini keterangan diagram relasional entitas bisnis :

1. Karyawan adalah direktur, *manager*, *assembling*, *machining*, *engineering*, dan administrasi.
2. Direktur adalah sebagai pemilik dan *surveyor*.
3. *Manager* diarahkan oleh direktur. *Manager* adalah sebagai *marketing* divisi teknik. *Manager* Mengatur *manager*, *assembling*, *machining*, *engineering*, dan administrasi.
4. *Machining* bertanggung jawab untuk pemesanan bahan baku ke *vendor* dan membuat turbin.
5. *Engineering* bertanggung jawab mendesain turbin.
6. *Assembling* bertanggung jawab merakit dan mencat turbin.
7. Administrasi bertanggung jawab dalam melayani karyawan.

3.2. Kolom HOW

Bagian yang diuraikan pada kolom *How* yaitu *scope*, *enterprise model*, dan *system model*. Berikut ini penjelasan *scope*, *enterprise model*, dan *system model*.

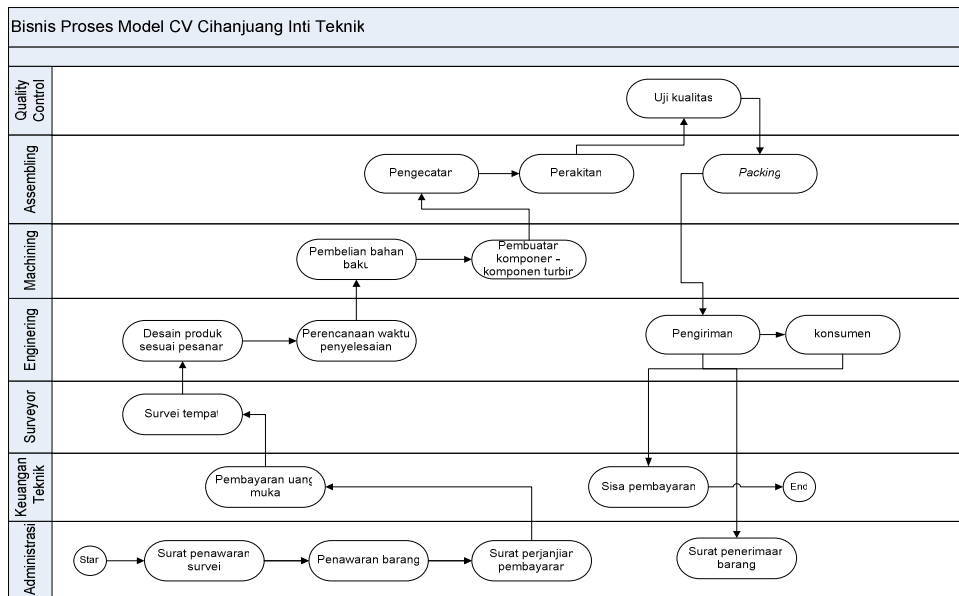
3.2.1. SCOPE : PLANNER PERSPECTIVE

Pada bagian *scope* membahas mengenai proses – proses utama yang saat ini terjadi di CV Cihanjuang Inti Teknik. Berikut ini penjelasan dari proses tersebut:

1. Proses penawaran
2. Survei
3. Pembuatan turbin
4. Pengiriman turbin

3.2.2. ENTERPRISE MODEL : OWNER PERSPECTIVE

Pada bagian *enterprise model* menggambarkan bisnis proses CV Cihanjuang Inti Teknik saat ini. Berikut ini merupakan gambar proses bisnis yang digambarkan menggunakan *functional flow diagram*.



Gambar III.2 Flow proses bisnis

3.3. Kolom *WHERE*

Bagian yang diuraikan pada kolom *Where* yaitu *scope*, *enterprise model*, *system model* dan *technology model*. Kolom *Where* membahas mengenai lokasi bisnis CV Cihanjuang Inti Teknik dalam menjalankan aktivitas bisnis saat ini. Berikut ini penjelasan *scope*, *enterprise model*, *system model* dan *technology model*.

3.3.1. SCOPE : PLANNER PERSPECTIVE

Pada bagian *scope* diuraikan mengenai tempat atau lokasi dari CV cihanjuang Inti teknik saat ini.

Berikut daftar lokasi bisnis CV Cihanjuang Inti Teknik:

1. Kantor : Jl. Cihanjuang 204 Cimahi Utara 40513 Kota Cimahi – Jawa Barat Indonesia.
2. Laboratorium pengujian : Memanfaatkan saluran irigasi sungai Leuwi Layung kampung Babut Girang - Cimahi Utara

3.3.2. ENTERPRISE MODEL : OWNER PERSPECTIVE

Pada bagian *enterprise model* menggambarkan lokasi bisnis dan peta lokasi CV Cihanjuang Inti Teknik.

3.4. Kolom *WHO*

Bagian yang diuraikan pada kolom *Who* yaitu *scope*, *enterprise model*, *system model*, *technology model* dan *functioning system*. Berikut ini penjelasan *scope*, *enterprise model*, *system model*, *technology model*, dan *functioning system*.

3.4.1. SCOPE : PLANNER PERSPECTIVE

Pada bagian *scope* menguraikan unit organisasi yang ada pada CV Cihanjuang Inti Teknik saat ini.

1. Direktur CV Cihanjuang Inti Teknik

2. *Manager CV Cihanjuang Inti Teknik*
3. *Administrasi Umum CV Cihanjuang Inti Teknik*
4. *Divisi Teknik CV Cihanjuang Inti Teknik* terdiri dari :
 - a. *Keuangan Teknik*
 - b. *Surveyor*
 - c. *Engineering*
 - d. *Machining*
 - e. *Assembling*
 - f. *Quality Control*
 - g. *Laboratorium Pengujian*
 - h. *Control*
 - i. *Pelatihan*
 - j. *Riset*
 - k. *Marketing*
5. *Divisi Makanan dan Minuman CV Cihanjuang Inti Teknik* terdiri dari :
 - a. *Keuangan Makanan dan Minuman*
 - b. *Produksi*
 - c. *Quality Control*
 - d. *Gudang*
 - e. *Marketing*

3.4.2. ENTERPRISE MODEL : OWNER PERSPECTIVE

Pada bagian *enterprise model* digambarkan kerangka struktur organisasi CV Cihanjuang Inti Teknik.

3.4.3. SYSTEM MODEL : ARCHITECT PERSPECTIVE

Pada bagian *system model* membahas peran dari setiap unit organisasi yang ada pada CV Cihanjuang Inti Teknik saat ini.

Tabel III.1 Peranan dari setiap Unit Organisasi

No	Unit Organisasi	Peranan
1	Direktur	Penanggung jawab seluruh kegiatan bisnis yang ada pada perusahaan.
2	<i>Manager</i>	Bertindak sebagai wakil direktur.
3	Administrasi Umum	Mengatur seluruh administrasi dan tata usaha perusahaan.
4	Keuangan Teknik	Mengatur seluruh alur keuangan yang ada pada divisi teknik.
5	<i>Surveyor</i>	Peninjau lapangan.
6	<i>Machining</i>	Pelaksana proses produksi turbin.
7	<i>Assembling</i>	Pelaksana assembler pada turbin.
8	<i>Quality Control (Teknik)</i>	Pengecekan produk turbin.
9	Laboratorium Pengujian	Tempat pengujian turbin
10	<i>Control (Teknik)</i>	Mengontrol sistem teknik dan service.
11	Pelatihan (Teknik)	Pelatihan untuk operator.
12	Riset (Teknik)	Pengembang produk.
13	Keuangan MakMin	Mengatur seluruh alur keuangan yang ada pada divisi Makanan dan Minuman.
14	Produksi (MakMin)	Pelaksana proses produksi minuman
15	<i>Quality Control (Makmin)</i>	Pengecekan produk minuman.
16	Gudang (MakMin)	Penyimpanan hasil produksi minuman.
17	<i>Marketing</i>	Pendistribusian produk kepada konsumen.

3.4.5. FUNCTIONING SYSTEM : USER PERSPECTIVE

Pada bagian *functional system* digambarkan struktuk organisasi CV Cihanjuang Inti Teknik saat ini.

3.5. Kolom WHEN

Bagian yang diuraikan pada kolom *When* yaitu *scope*, *enterprise model*, *system model*, *technology model*, dan *functioning system*. Berikut ini penjelasan *scope*, *enterprise model*, *system model*, *technology model*, dan *functioning system*.

3.5.1. SCOPE : PLANNER PERSPECTIVE

Pada bagian *scope* membahas mengenai daftar kegiatan utama yang dilaksanakan CV Cihanjuang Inti Teknik saat ini. Berikut ini uraian kegiatan bisnis CV Cihanjuang Inti Teknik:

1. Mengikuti Pameran yang diselenggarakan Dinas Pertambangan dan Departemen Perindustrian dan Perdagangan.
2. Perekrutan karyawan
3. Rapat karyawan
4. Permintaan bahan baku produksi
5. Pengontrolan keuangan
6. Penggajian karyawan

3.5.2. ENTERPRISE MODEL : OWNER PERSPECTIVE

Pada bagian *enterprise model* akan dijelaskan lebih rinci mengenai pelaksanaan kegiatan bisnis CV Cihanjuang Inti Teknik saat ini.

3.5.3. SYSTEM MODEL : ARCHITECTURE PERSPECTIVE

Pada bagian *system model* menguraikan pengaturan waktu berdasarkan jangka waktu tertentu untuk setiap pelaksanaan kegiatan bisnis CV Cihanjuang Inti Teknik saat ini.

3.5.4. FUNCTIONING SYSTEM : USER PERSPECTIVE

Pada bagian *fuctioning system* diuraikan penjadwalan pelaksanaan kegiatan bisnis CV Cihanjuang Inti Teknik berdasarkan kalender saat ini.

3.6. Kolom WHY

Bagian yang diuraikan pada kolom *Why* yaitu *scope* dan *enterprise model*. Berikut ini penjelasan *scope* dan *enterprise model*.

3.6.1. SCOPE : PLANNER PERSPECTIVE

Teknologi mikrohidro adalah pilihan CV. CIHANJUANG INTI TEKNIK (CIT) atau (CINTEK) untuk memberdayakan ekonomi masyarakat. Berikut ini visi, misi CIT saat ini:

3.6.1.1. VISI

1. Energi sebagai sumber pemberdayaan ekonomi masyarakat
2. Manfaatkan potensi alam yang terabaikan
3. Mendorong upaya penyelamatan lingkungan

4. Pengembangan Teknologi Bisnis.

3.6.1.2. MISI

1. Memberikan hasil produk yang dapat dimanfaatkan masyarakat
2. Produk Hanjuang sebagai penggerak ekonomi pedesaan
3. Dapat dijadikan sebagai tempat penelitian untuk kemajuan pendidikan di Indonesia.

3.6.2. ENTERPRISE MODEL : OWNER PERSPECTIVE

Pada bagian *enterprise model* akan diuraikan mengenai perencanaan dari CV Cihanjuang Inti Teknik dalam jangka panjang yang digambarkan dalam *Balance Score Card* dan teknik analisis bisnis yang digunakan adalah SWOT Analisis.

III.6.2.1. SWOT Analisis

SWOT analisis membahas mengenai kekuatan, kelemahan, kesempatan serta tantangan yang dimiliki CV Cihanjuang Inti Teknik saat ini.

Tabel III.2 SWOT analisis

<ul style="list-style-type: none"> • Kolom <i>What</i> 	
<p>Strengths :</p> <ol style="list-style-type: none"> 1. Mempunyai hak paten dalam perizinan pendirian usaha CV Cihanjuang Inti Teknik. 2. Perusahaan saat ini memiliki dua macam bidang produksi yang berbeda. 	<p>Weakness :</p> <ol style="list-style-type: none"> 1. Saat ini pengelolaan data perusahaan masih dilakukan secara manual. Contohnya pencatatan perumusan turbin masih dalam bentuk dokumen, tidak ada sistem yang khusus menangani perhitungan rumus turbin. 2. Perlu adanya tempat khusus untuk menyimpan dokumen – dokumen perusahaan. Seperti loker penyimpanan <i>file</i>. 3. Data perusahaan tidak dapat di <i>share</i> ke setiap divisi. Contohnya jika direktur ingin mengetahui pendapatan perusahaan bulan lalu, harus mencari dokumen keuangan bulan lalu terlebih dahulu.
<p>Opportunities :</p> <ol style="list-style-type: none"> 1. Adanya pengembangan produk baru yang sangat bermanfaat bagi masyarakat dan ramah lingkungan. Contohnya pemanfaatan angin sebagai sumber energi listrik, dengan membuat kincir angin . 2. Terbukanya kesempatan untuk menambah bidang produksi. Contohnya pengolahan sampah yang dapat didaur ulang seperti 	<p>Threats :</p> <ol style="list-style-type: none"> 1. Resiko kehilangan data – data penting perusahaan. Contohnya data survei tempat pemasangan turbin. 2. Adanya resiko kerusakan terhadap data penting perusahaan. Contohnya dokumen berjamur, dikarenakan tempat penyimpanan yang lembab. 3. Kemungkinan adanya redundansi data. Contohnya pencatatan data konsumen yang sudah pernah melakukan transaksi bisnis dengan perusahaan, tetapi karena lama tidak melakukan transaksi dan membutuhkan waktu untuk mencari

plastik. 3. Pengolahan data yang dilakukan oleh sistem. Contohnya sistem penanganan perhitungan turbin.	data tersebut sehingga cara yang mudah adalah mencatat kembali. 4. Data tidak konsisten. Karena adanya kemungkinan redundansi data, maka sulit untuk menentukan data yang valid.
--	---

• **Kolom How**

Strengths : <ol style="list-style-type: none"> 1. Setiap proses yang ada di perusahaan sudah tepat, dengan adanya pembagian kerja di setiap divisi. 2. Alur proses bisnis yang ada di perusahaan sudah jelas. 	Weakness : <ol style="list-style-type: none"> 1. Perusahaan saat ini tidak memiliki sistem yang menangani setiap proses bisnis yang ada di perusahaan. Contohnya pencatatan data keuangan. 2. Aliran data di perusahaan tidak dapat digambarkan. Karena tidak ada basis data.
Opportunities : <ol style="list-style-type: none"> 1. Menggunakan sistem yang dapat menangani proses bisnis yang ada di perusahaan. Contohnya adanya basis data yang dapat membantu dalam penyimpanan dan pencarian data yang akhirnya dapat digunakan dalam pengambilan keputusan. 	Threats : <ol style="list-style-type: none"> 1. Pengolahan data tidak efektif dan efisien. Contohnya kemungkinan kesalahan manusia dalam mencatat data dan menghitung biaya. 2. Sumber daya perusahaan tidak dapat digunakan dengan sebaik – baiknya. Contohnya dalam pencatatan absensi karyawan yang berguna untuk perhitungan gaji yang seharusnya dapat diolah oleh sistem, tetapi harus dilakukan oleh karyawan.

• **Kolom Where**

Strengths : <ol style="list-style-type: none"> 1. Pemilihan lokasi bisnis jaraknya tidak jauh dari tempat laboratorium pengujian. Sehingga pengecekan kualitas produk tidak menghabiskan waktu dan biaya. 2. Peta lokasi bisnis dan ruang kerja telah tergambar dengan jelas. Sehingga orang luar dapat mengetahui posisi dan lokasi perusahaan. 	Weakness : <ol style="list-style-type: none"> 1. Jarak antara ruang kerja dan pabrik berdekatan, sehingga adanya polusi suara. Hal ini dapat dilihat dari denah ruang kerja. 2. Luas pabrik tidak memadai untuk proses produksi. Hal ini dilihat dari luas perusahaan lebih kurang 500m². 3. Perusahaan tidak memiliki jaringan untuk kegiatan bisnis yang menghubungkan setiap komputer yang ada di perusahaan.
Opportunities : <ol style="list-style-type: none"> 1. Membuat suatu jaringan yang dapat menghubungkan semua komputer yang ada di setiap divisi, untuk 	Threats : <ol style="list-style-type: none"> 1. Luas area pabrik yang tidak mencukupi untuk produksi barang dan penempatan produk dalam skala besar. Hal ini dapat berpengaruh terhadap kepuasan konsumen,

<p>kelancaran proses bisnis. Contohnya komputer yang ada di ruang keuangan dan ruang divisi teknik dapat terhubung sehingga data dapat di <i>share</i>.</p> <p>2. Melakukan penataan kembali terhadap ruang kerja. Karena adanya polusi suara.</p> <p>3. Membuka cabang bisnis baru. Hal ini dilihat dari strategi perusahaan.</p>	<p>karena jika perusahaan mendapat permintaan dalam jumlah besar tidak dapat terpenuhi sehingga konsumen beralih ke perusahaan lain yang dapat memenuhi permintaan mereka.</p>
--	--

• Kolom *Who*

<p>Strengths :</p> <p>1. loyalitas karyawan tinggi. Karena saat ini jumlah karyawan lebih kurang 100 orang yang rata – rata telah bekerja lebih dari lima tahun.</p> <p>2. Adanya hubungan baik antara atasan dan bawahan. Hal ini dilihat dari kesedian atasan untuk membantu mengajari karyawan.</p>	<p>Weakness :</p> <p>1. Tenaga kerja kurang sehingga tidak dapat menangani permintaan pasar dalam partai besar.</p> <p>2. Masih ada peran tenaga kerja yang merangkap sehingga kurang fokus dalam menangani kegiatan bisnis. Contohnya <i>manager</i> dan <i>marketing</i> divisi teknik di jabat oleh orang yang sama.</p>
<p>Opportunities :</p> <p>1. Membuka lapangan pekerjaan bagi masyarakat. Contohnya jika permintaan dari konsumen dalam partai besar, pihak perusahaan harus merekrut karyawan baru agar produksi dapat berjalan lancar dan tepat waktu.</p> <p>2. Mencari tenaga ahli di bidang survei. Contohnya mengajari karyawan yang sudah lama mengabdikan pada perusahaan untuk menjadi asisten dalam melakukan survei.</p>	<p>Threats :</p> <p>1. Konsumen kecewa karena permintaan yang diajukan, tidak dapat terpenuhi oleh perusahaan. Karena keterbatasan sumber daya manusia.</p> <p>2. Kurangnya para ahli dalam penanganan masalah – masalah tertentu yang ada pada perusahaan seperti orang yang dapat melakukan survei tempat pemasangan turbin.</p>

• Kolom *When*

<p>Strengths :</p> <p>2. Harga jual produk dapat dijangkau oleh masyarakat pada umumnya. Hal ini dilihat dari visi, yaitu energi sebagai sumber pemberdayaan ekonomi</p>	<p>Weakness :</p> <p>2. Penggajian tenaga kerja masih dilakukan secara manual. Penerimaan uang gaji tidak melalui bank.</p> <p>3. Adanya kemungkinan salah dalam menghitung gaji karyawan. Karena perhitungan gaji tidak dilakukan secara otomatis oleh sistem, tetapi</p>
---	---

<p>masyarakat, dan misi perusahaan, yaitu produk Hanjuang sebagai penggerak ekonomi pedesaan.</p> <p>3. Memproduksi produk yang ramah lingkungan dan dapat digunakan untuk mengelola alam. Hal ini dilihat dari visi perusahaan, yaitu memanfaatkan potensi alam yang terabaikan, dan Mendorong upaya penyelamatan lingkungan.</p>	<p>dilakukan oleh manusia.</p> <p>4. Perusahaan saat ini belum bisa memenuhi permintaan untuk mengikuti pameran industri diluar negeri. Karena masih merupakan rencana jangka panjang perusahaan.</p>
<p>Opportunities :</p> <ol style="list-style-type: none"> 1. Melakukan kerja sama dengan bank tertentu untuk proses transaksi bisnis yang dilakukan perusahaan. Contohnya dalam pengiriman gaji karyawan. 2. Membuka kerja sama dengan SMK untuk pembuatan produk sehingga saling menguntungkan antar kedua belah pihak. Hal ini dapat dilihat dari misi perusahaan yaitu dapat dijadikan tempat penelitian untuk kemajuan pendidikan di Indonesia. 3. Pesaing bisnis khususnya dalam pembuatan turbin masih sedikit. Sehingga perusahaan mempunyai startegi untuk membuka cabang perusahaan baru. 	<p>Threats :</p> <ol style="list-style-type: none"> 1. Pemasaran turbin saat ini masih terbatas, sebagian besar hanya wilayah Indonesia. Saat ini perusahaan belum dapat memasarkan ke luar negeri. 2. <i>Vendor</i> untuk pemesanan bahan baku turbin masih terbatas.

5. Kesimpulan dan Saran

Pendokumentasian sangat penting dilakukan oleh setiap perusahaan, karena dengan adanya pendefinisian keadaan perusahaan secara jelas dapat membantu proses pengembangan perusahaan dimasa yang akan datang. Dengan demikian perusahaan dapat mengetahui sejauh mana perkembangan perusahaan saat ini. Dokumen *Enterprise Architecture* ini dapat digunakan untuk mengembangkan sistem komputerisasi yang dibutuhkan oleh CV Cihanjuang Inti Teknik untuk meningkatkan efektifitas dan efisiensi kinerjanya.

Pada saat ini, penulis mencapai tahap pendefinisian organisasi dimana data diperoleh dari hasil wawancara dan analisis organisasi. Penyusunan dokumen, masih belum sempurna karena masih terdapat kolom yang belum terimplementasi. Saran penulis bagi pengembang selanjutnya dapat mengimplementasikan seluruh kolom yang ada pada *Zachman framework* dan data – data yang telah didokumentasi dapat terus diperbaharui.

6. DAFTAR PUSTAKA

- [1] Zachman Institute for Framework Advancement : The Framework for Enterprise Architecture. Retrieved : January, 2008, from <http://www.zifa.com/>
- [2] Hay, David. C. (1997). The Zachman Framework : An Introduction. Retrieved : January, 2008, from <http://www.tdan.com/view-articles/>
- [3] Zachman Framework Applied to Administrative Computing. Retrieved : January, 2008, from [http:// www.zifa.com/Zachman/](http://www.zifa.com/Zachman/)
- [4] Spewak, Steven H., Hill, Steven C. (1992). Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology. New York: Jon Wiley & Sons, Inc.
- [5] O'Rourke, Carol., Fishman, Neal., Selkow, Warenn. (2003). Enterprise Architecture the Zachman Framework. Boston Massachusetts: Thomson Course Technology.
- [6] Institute for Enterprise Architecture Development: Enterprise Architecture Definition. Retrieved: April, 2008, from [http:// http://www.enterprise-architecture.info/](http://www.enterprise-architecture.info/)
- [7] Imbar, Radiant. V. (2007). Diktat Mata Kuliah Pemodelan Sistem Informasi.
- [8] The Open Group.(2006).Other Architectures and Frameworks. Retrieved: April, 2008, from [http:// www. opengroup. org/ architecture/togaf8-doc/arch/](http://www.opengroup.org/architecture/togaf8-doc/arch/)
- [9] Imbar, Radiant. V., Christianti, Meliana. (2006). Pemodelan Enterprise Architecture Zachman Framework pada Sistem Informasi Fakultas Teknologi Universitas Kristen Maranatha Bandung. Bandung : Jurnal Sistem Informasi. (Vol. 2 / No. 2/ September 2007, p. 113 - 135).
- [10] The Zachman Framework and the OMG's Model Driven Architecture Retrieved: April, 2008. from http://www.omg.org/mda/mda_files/

Perangkat Lunak JIT (Just In Time) untuk Memprediksi Resiko Proyek Perangkat Lunak

Yasmi Afrizal^{1*}, Agus Harjoko²

¹Universitas Komputer Indonesia, Bandung

²Universitas Gadjah Mada, Yogyakarta

email: yasmi_afrizal@yahoo.co.id¹, aharjoko@ugm.ac.id²

Abstract

Managing software projects can often degrade into fighting fires lit by the embers of unrecognized and unmanaged risks. Stakeholders are a recognized source of significant software project risk, but few researchers have focused on providing a practical method for identifying specific software project risk.

We used software JIT(Just In Time) model to provide this practical guidance. Software JIT offers the project team a step-by-step approach to identifying risks during requirements engineering on the project and assessing the risks that their potential negative responses pose. We illustrate software JIT using a case study of a simulated akademik project that aims to reduce the risks software development.

Keywords : Software project, Just In Time, risk,

I. PENDAHULUAN

Industri perangkat lunak telah menjadi salah faktor penentu pertumbuhan ekonomi dunia. Perkembangan tersebut tidak lepas dari kebutuhan masyarakat terhadap teknologi informasi dan komunikasi yang semakin meningkat. Pada tahun 2007, pertumbuhan industri perangkat lunak Indonesia telah mencapai sekitar 56.500 pengembang dan terus akan bertambah sampai 71.600 pengembang tahun 2008 [10]. Pertumbuhan yang cepat tersebut, membuat pengembang perangkat lunak semakin berkompetisi untuk mengambil lompatan-lompatan besar dalam mengembangkan perangkat lunak mereka. Akibat permasalahan di atas mendorong pengembang atau dunia industri untuk mengambil resiko lebih besar. Dunia industri dan masyarakat perangkat lunak harus sadar akan pentingnya manajemen resiko dan beberapa pendekatan untuk diterapkan dalam proyek mereka.

Tujuan utama manajemen resiko adalah mengenali semua kemungkinan kegagalan dari suatu proyek perangkat lunak dengan melihat kekomplekan dalam memutuskan langkah solusi yang akan dibuat secara alami [1]. Solusi pemecahan masalah dilakukan dengan meminimalkan segala macam ketidakjelasan yang muncul dan melakukan evaluasi terhadap pemecahan tersebut.

Perangkat Lunak JIT (*Just In Time*) adalah salah satu metode yang digunakan untuk melakukan mengelola resiko dalam pembangunan perangkat lunak, dimana kunci pendekatannya adalah melakukan perencanaan di awal proyek [6]. Pada manajemen resiko perangkat lunak perencanaan hampir semua mengarah pada segi bisnis. Hasil perencanaan menyatakan tujuan akhir (*goal*) yang nantinya dimasukan ke dalam persyaratan perangkat lunak.

Sebelum penelitian ini dilakukan, Konsep JIT telah banyak diterapkan pada dunia industri, terutama dalam menilai dan mengevaluasi rancangan, aliran dan jadual dari produk yang dihasilkan, dengan mengeliminasi waktu yang terbuang, sehingga semua aktivitas yang dilakukan dapat menambah nilai bagi perusahaan.

Pengertian perangkat lunak JIT, memiliki pengertian yang sama dengan JIT industri yaitu melakukan perencanaan untuk menilai dan mengevaluasi produk yang dihasilkan di awal proyek.

Terdapat 2 (dua) alasan, mengapa penelitian ini menggunakan JIT sebagai pendekatan dalam mengelola resiko. Pertama, JIT merupakan salah satu metode yang paling sukses diterapkan pada seluruh aktivitas proyek industri. Kedua, perangkat lunak JIT dapat mengenali seluruh ruang lingkup resiko proyek perangkat lunak, sehingga penilaian dan evaluasi resiko yang dilakukan dapat menjamin kehandalan dari perangkat lunak yang dihasilkan.

Pada manajemen resiko perangkat lunak, hampir semua tahap perencanaan mengarah pada segi bisnis. Hasil perencanaan menyatakan tujuan akhir (*goal*) yang nantinya dimasukkan ke dalam persyaratan perangkat lunak.

2. RUANG LINGKUP PERANGKAT LUNAK JIT (JUST IN TIME)

Beberapa penelitian mengenai manajemen resiko telah diperkenalkan dan dikembangkan oleh beberapa peneliti. Kumpulan penelitian tersebut tidak dapat kita perbandingkan antara satu dengan yang lainnya, disebabkan ruang lingkup penelitian manajemen resiko yang digunakan berbeda-beda. Manajemen resiko perangkat lunak harus dapat dianalisis, dinilai dan dievaluasi dari berbagai ruang lingkup proyek. Pada perangkat lunak JIT ruang lingkup manajemen resiko terdiri dari : elemen resiko, aktivitas resiko, faktor resiko, matrik resiko dan metodologi life cycle.

2.1 Element Resiko

Penerapan manajemen resiko pada proyek perangkat lunak tidak lepas dari pertimbangan teknologi dan bisnis. Perspektif teknologi menjelaskan alat bantu (*tools*), teknik dan lingkungan, dimana perangkat lunak tersebut diterapkan. Perspektif bisnis menjelaskan sumber daya, jadwal dan dampak bisnis (keberhasilan pembangunan perangkat lunak).

Perangkat lunak JIT mampu untuk mengelola resiko perangkat lunak, baik menurut perspektif teknologi maupun bisnis[6]. Tidak semua resiko dalam perspektif di atas masuk ke dalam resiko perangkat lunak. Hanya terdapat 3 elemen dari resiko yang digunakan dalam perangkat lunak JIT yaitu teknologi, biaya dan penjadualan.

Elemen teknologi berhubungan dengan kinerja perangkat lunak, yaitu : kehandalan, kualitas, fungsi, pemeliharaan dan kegunaan kembali. Elemen biaya berhubungan dengan biaya perangkat lunak selama pembangunan perangkat lunak seperti *variable cost*, *fix cost* dan *budget*. Sedangkan elemen penjadualan berhubungan dengan jadwal proyek selama membangun perangkat lunak, seperti : jadwal realisasi, jadwal pertemuan dengan pelanggan dan anggota pengembang dan jadwal perubahan waktu proyek.

2.2. Aktivitas Resiko

Aktivitas resiko merupakan cara melakukan evaluasi terhadap resiko berdasarkan pandangan dari operasional, strategi, teknologi, bisnis, industri dan para praktisi [6]. Terdapat 6 (enam) aktivitas yang dilakukan dalam mengevaluasi manajemen resiko perangkat lunak yaitu :

1. Identifikasi resiko yaitu melakukan pengumpulan informasi mengenai proyek perangkat lunak dan mengklasifikasikan informasi tersebut untuk menentukan resiko yang paling potensial dari suatu proyek. Informasi dikumpulkan dengan merujuk data pada proyek perangkat lunak yang pernah dikerjakan.
2. Strategi dan perencanaan resiko yaitu mengembangkan alternatif-alternatif resiko yang akan muncul selama pembangunan perangkat lunak.
3. Penilaian resiko adalah memutuskan dampak resiko yang paling potensial melalui suatu penilaian.
4. Pengurangan/penghindaran resiko yaitu aktivitas yang dilakukan dalam meminimalkan atau menghindari efek resiko
5. Membuat laporan digunakan untuk mendokumentasikan pengelolaan resiko dari proyek perangkat lunak, termasuk melakukan perbandingan status resiko dengan resiko proyek yang pernah dikerjakan
6. Prediksi resiko yaitu melakukan prediksi tentang perkembangan resiko dari proyek dengan menggunakan interaksi data dan pengetahuan

2.3. Faktor Resiko Perangkat Lunak

Meskipun secara tidak langsung berpengaruh terhadap perangkat lunak. Faktor resiko sangat bermanfaat dalam menjelaskan karakteristik proyek yang dikerjakan pada masa lalu. Penelitian dari Mc Call dan Boehm [1][2][8] menjelaskan terdapat 10 faktor resiko perangkat lunak, dimana faktor resiko tersebut berhubungan dengan kualitas dan kehandalan produk perangkat lunak.

Satu faktor resiko dapat berhubungan lebih dari satu elemen resiko. Faktor resiko juga berpengaruh terhadap proses dan produk perangkat lunak. Berdasarkan pengalaman industri perangkat lunak, setiap faktor resiko diberi pembobotan penilaian berupa tinggi, sedang dan rendah seperti terlihat pada tabel 1, dimana bobot tersebut menyatakan derajat pengaruh faktor resiko terhadap elemen resiko.

Tabel 1. Derajat pengaruh faktor resiko terhadap elemen resiko

Faktor Resiko	Elemen resiko perangkat lunak		
	Teknologi	Biaya	Penjadualan
<i>Organization</i>	rendah	tinggi	tinggi
<i>Estimation</i>	rendah	tinggi	tinggi
<i>Monitoring</i>	sedang	tinggi	tinggi
<i>Development Methodology</i>	sedang	tinggi	tinggi
<i>Tools</i>	sedang	sedang	sedang
<i>Risk Culture</i>	tinggi	sedang	sedang
<i>Usability</i>	tinggi	rendah	rendah
<i>Correctness</i>	tinggi	rendah	rendah
<i>Reability</i>	tinggi	rendah	rendah
<i>Personel</i>	tinggi	tinggi	tinggi

2.4. Matrik Resiko

Matrik resiko digunakan untuk menilai faktor resiko dalam perangkat lunak. Konsep ini dikemukakan pertama kali oleh Mc Call dan Boehm [1][2][8] yang berfungsi untuk mendapatkan perangkat lunak yang berkualitas dan handal. Matrik

resiko perangkat lunak merupakan kumpulan pertanyaan (kuesioner) dengan jawaban yang diberi bobot nilai sesuai dengan pendapat para manajemen proyek perangkat lunak.

2.5. Metodologi Just In Time

Metodologi manajemen resiko menjelaskan langkah atau aktivitas yang diambil untuk mengelola resiko pada setiap phase model proses pengembangan perangkat lunak *life cycle*. Metodologi JIT menghubungkan model proses dengan aktivitas evaluasi manajemen resiko.

Aktivitas evaluasi resiko yang terdiri dari : identifikasi, strategi dan perencanaan, penilaian, pengurangan/penghindaran, laporan dan prediksi yang kemudian dijabarkan pada setiap tahapan (*phase*) model proses *life cycle* [1][2][8]. Setiap phase *life cycle* dievaluasi menggunakan pertanyaan dari matrik resiko yang merujuk 10 (sepuluh) faktor resiko yaitu : *organization, estimation, monitoring, development methodology, tool, risk culture, correctness, reliability* dan *personel*.

Pada metodologi *Just In Time*, tidak semua pertanyaan pada matrik resiko diterapkan pada setiap phase *life cycle*. Hal ini disebabkan setiap pertanyaan memiliki karakteristik sendiri dalam menilai faktor resiko yang ada pada setiap phase *life cycle*.

2.6. Desain Model SERIM

The Software Engineering Risk Model (SERIM) merupakan model yang digunakan JIT untuk memberikan manajemen pemecahan alternatif resiko pada suatu proyek perangkat lunak. SERIM digunakan sebagai pendekatan untuk menghitung resiko perangkat lunak. Pendekatan berdasarkan subyek-subyek kemungkinan berdasarkan pengalaman dan analogi kejadian. Jika terdapat dua kejadian yaitu A dan B, dimana kemungkinan $P(A)$ lebih besar dari kemungkinan $P(B)$, maka kemungkinan $P(A)$ akan lebih mendapatkan perhatian dari kemungkinan $P(B)$. SERIM dihitung berdasarkan pertanyaan matrik resiko perangkat lunak. Nilai yang didapatkan dari pertanyaan tersebut akan selalu berbeda antara satu orang dengan yang lain, hal ini disebabkan jawaban didasarkan pada perasaan setiap orang dari pengalaman masing-masing terhadap produk bisnis dan lingkungan dalam membangun perangkat lunak.

3. STUDI KASUS

Pada sesi ini, peneliti mencoba untuk menerapkan perangkat lunak JIT dalam proyek perangkat lunak, dimana penelitian dilakukan Universitas Komputer Indonesia Bandung pada unit UNIKOM Center. Tugas utama unit UNIKOM center adalah mengembangkan seluruh perangkat lunak berada di bawah lingkungan universitas. Pada tahun 2009 UNIKOM center merencanakan untuk membangun sistem akademik yang terpadu, sehingga memungkinkan dosen, mahasiswa dan orang tua dapat berinteraksi dalam suatu wadah. Perangkat lunak tersebut, direncanakan dibangun selama 8 bulan dengan mempekerjakan 8 programmer, 1 sistem analis dengan bahasa pemrograman dasar PHP dan java.

Untuk menghindari kegagalan dalam proyek tersebut, peneliti dan pihak UNIKOM center merencanakan melakukan analisis resiko pada tahap awal pembangunan perangkat lunak. Adapun tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Menerapkan perangkat lunak JIT dalam proyek, sehingga dapat mengetahui ruang lingkup resiko proyek dan strategi melakukan analisis dan penilaian terhadap resiko yang muncul.
2. Memperbaiki operasi dari perangkat lunak pada organisasi dan memberikan kontribusi untuk mengembangkan pengetahuan tentang bagaimana merancang, mengatur dan mengelola resiko pada proyek perangkat lunak

4. PENGUMPULAN DATA

Penggunaan kuisioner merupakan salah satu metode juga paling populer dan luas untuk mengumpulkan data dan informasi [7]. Kuesioner matrik resiko yang digunakan pada penelitian, terdiri 81 pertanyaan yang berfungsi melakukan evaluasi terhadap ruang lingkup perangkat lunak JIT. Sedangkan analisis dan programmer merupakan representatif responden dari informasi yang dikumpulkan. Pertanyaan yang digunakan dalam penelitian ini dapat dilihat pada lampiran dengan contoh pertanyaan sebagai berikut :

1. Apakah anda melakukan perancangan membangun perangkat lunak berdasarkan pengalaman perusahaan anda dalam mengerjakan proyek perangkat lunak sebelumnya ?
2. Apakah terdapat komunikasi yang baik antara tim, jika terjadi perbedaan pendapat dalam organisasi untuk membangun perangkat lunak ?
3. Apakah estimasi biaya proyek didasarkan pada estimasi biaya proyek perangkat lunak sebelumnya ?

Secara umum bobot yang dibangkitkan untuk menjawab pertanyaan pada matrik resiko adalah nilai probabilitas antara 0 sampai 1 seperti yang terlihat pada tabel 2.

Tabel 2. Bobot nilai setiap jawaban kuisioner matrik resiko

Nilai Jawaban	Keterangan
0.0 – 0.2	Tidak pernah
0.2 - 0.4	Jarang
0.4 – 0.6	Kadang-kadang
0.6 – 0.8	Sering
0.8 – 1.0	Pasti

. Berdasarkan tabel di atas, maka kita dapat menjawab setiap pertanyaan matrik resiko dengan menggunakan nilai jawaban yang kita sesuaikan dengan derajat persetujuan pada kolom keterangan. Pertanyaan no 1. dari contoh, kita dapat memberikan nilai 0.17, jika perangkat lunak yang dibangun tidak berdasarkan pengalaman proyek sebelumnya. Nilai 0.3, jika pembangunan perangkat lunak jarang menggunakan pengalaman proyek sebelumnya. Nilai 0.5, jika pembangunan perangkat lunak kadang-kadang menggunakan pengalaman proyek sebelumnya. Nilai 0.65, jika pembangunan perangkat lunak sering menggunakan pengalaman proyek sebelumnya. Nilai 0.9, jika pembangunan perangkat lunak pasti menggunakan pengalaman proyek sebelumnya.

Pada tabel 3 memperlihatkan nilai respon pertanyaan matrik resiko yang diberikan kepada responden.

Tabel 3. Hasil respon pertanyaan matrik resiko

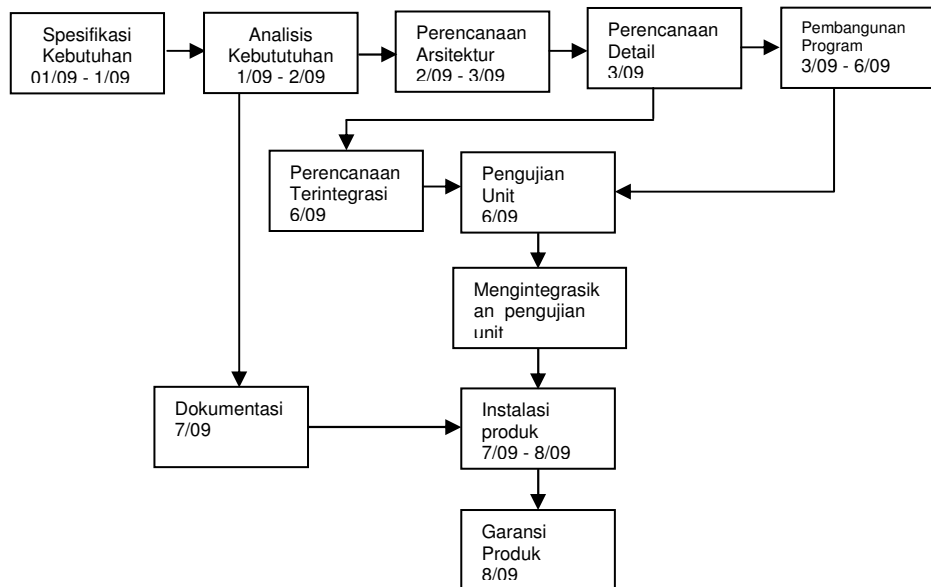
Q1 = 0.70	Q22 = 0.90	Q43 = 0.65	Q64 = 0.40
Q2 = 0.70	Q23 = 0.30	Q44 = 0.65	Q65 = 0.45
Q3 = 0.50	Q24 = 1.00	Q45 = 0.50	Q66 = 0.45
Q4 = 0.75	Q25 = 0.60	Q46 = 0.80	Q67 = 0.30
Q5 = 0.85	Q26 = 0.70	Q47 = 0.70	Q68 = 0.45
Q6 = 0.80	Q27 = 0.50	Q48 = 0.55	Q69 = 0.40
Q7 = 0.70	Q28 = 0.90	Q49 = 0.60	Q70 = 0.30
Q8 = 0.70	Q29 = 0.30	Q50 = 0.60	Q71 = 0.50
Q9 = 0.50	Q30 = 0.50	Q51 = 0.70	Q72 = 0.50
Q10 = 0.90	Q31 = 0.40	Q52 = 0.80	Q73 = 0.40
Q11 = 0.30	Q32 = 0.30	Q53 = 0.50	Q74 = 0.40
Q12 = 1.00	Q33 = 0.70	Q54 = 0.70	Q75 = 0.40
Q13 = 0.60	Q34 = 0.50	Q55 = 0.65	Q76 = 0.40
Q14 = 0.70	Q35 = 0.20	Q56 = 0.40	Q77 = 0.80
Q15 = 0.50	Q36 = 0.40	Q57 = 0.50	Q78 = 0.90
Q16 = 0.90	Q37 = 0.40	Q58 = 0.45	Q79 = 0.75
Q17 = 0.30	Q38 = 0.40	Q59 = 0.40	Q80 = 0.70
Q18 = 1.00	Q39 = 0.90	Q60 = 0.55	Q81 = 0.75
Q19 = 0.60	Q40 = 0.70	Q61 = 0.40	
Q20 = 0.70	Q41 = 0.57	Q62 = 0.55	
Q21 = 0.50	Q42 = 0.85	Q63 = 0.55	

Variabel Qn menunjukan pertanyaan (Q) dan nomor (n) dari matrik resiko, dimana n = 1,2,381. Bobot nilai pertanyaan diisi oleh responden sesuai dengan persetujuan dan pengalaman mereka masing-masing dalam membangun perangkat lunak.

5. ANALISIS DATA

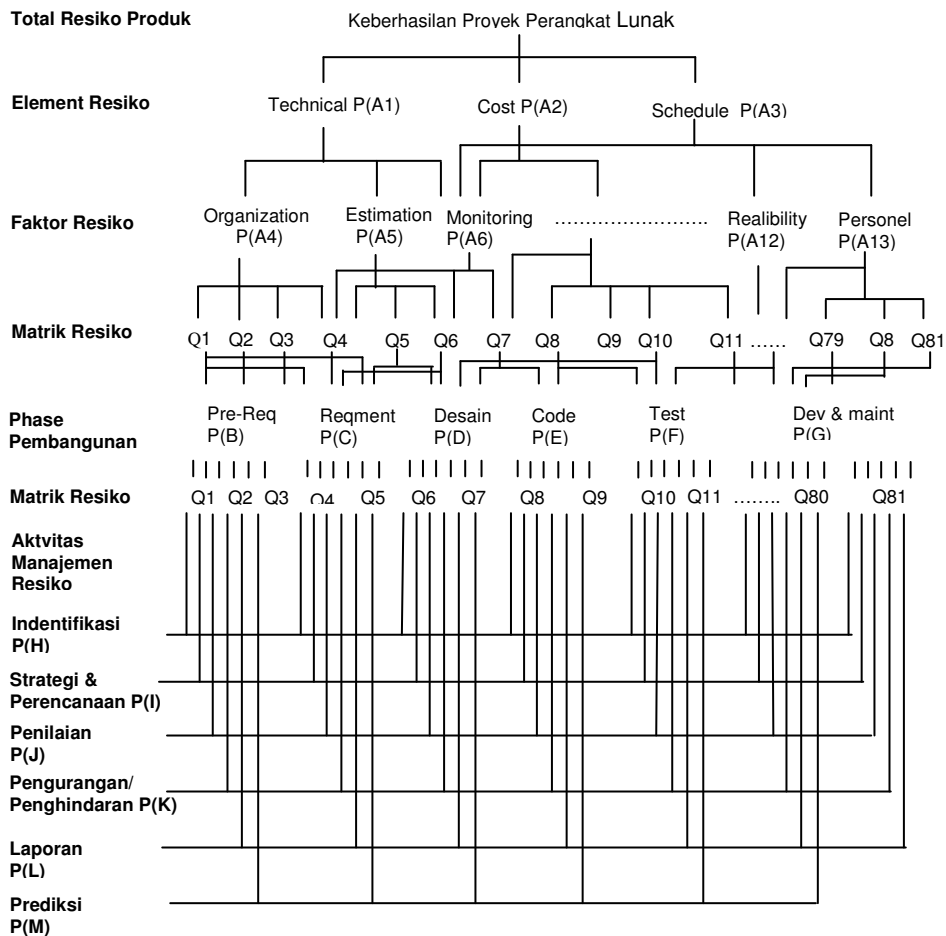
Sebelum analisis data dilakukan, diperlukan gambaran operasi yang memperlihatkan hubungan antara aktivitas-aktivitas yang ada dalam membangun perangkat lunak. Pondasi utama dari operasi perencanaan adalah bagaimana menangkap aktivitas setiap langkah pembangunan perangkat lunak, sehingga dapat digambarkan secara menyeluruh. Agar mendapatkan gambaran operasi tersebut diperlukan suatu alat bantu. *Program Evaluation and Review Technique* (PERT) merupakan alat bantu yang dapat menggambarkan peta jalan secara rinci setiap langkah pekerjaan dalam pembangunan perangkat lunak

Meskipun tidak digambarkan secara rinci, penelitian ini berusaha menggambarkan urutan langkah aktivitas yang akan dilakukan selama pembangunan perangkat lunak dengan format waktu bulan dan tahun pada setiap langkahnya seperti yang terlihat pada gambar 1.



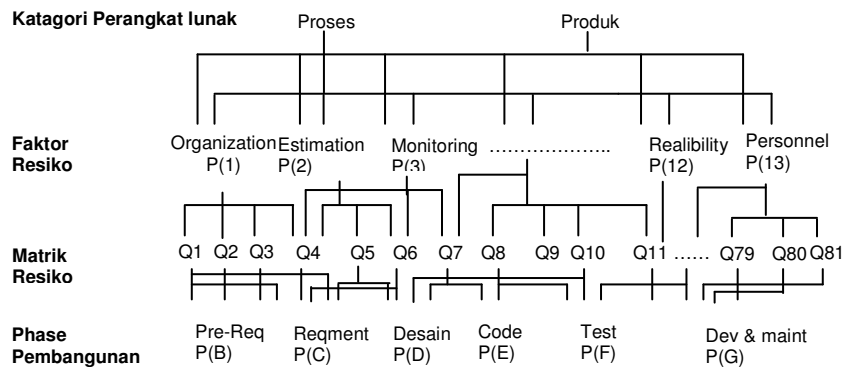
Gambar 1. Diagram PERT pembangunan perangkat lunak akademik

Sebelum proses penilaian terhadap perangkat lunak JIT dilakukan, maka diperlukan struktur pohon kemungkinan yang berfungsi menghubungkan perbedaan-perbedaan ruang lingkup yang ada pada perangkat lunak JIT Gambar 2. $P(A)$ menunjukkan total kesuksesan proyek perangkat lunak. $P(A_1)$, $P(A_2)$ dan $P(A_3)$ menjelaskan elemen resiko berupa teknologi, biaya dan penjadwalan. $P(A_4)$ sampai $P(A_{13})$ memperlihatkan 10 faktor resiko. $P(B)$ sampai $P(G)$ memperlihatkan tahapan *life cycle*, sedangkan $P(H)$ sampai $P(M)$ menjelaskan aktivitas dalam evaluasi resiko.



Gambar 2. Integrasi Model Manajemen Risiko
(sumber : software engineering risk management, dale walter karolak)

Gambar 3. Menjelaskan hubungan antara faktor resiko dengan proses pembangun dan produk perangkat lunak. P(N) menjelaskan kesuksesan proyek berdasarkan proses dan P(O) menjelaskan kemungkinan kesuksesan berdasarkan produk.



Gambar 3. Model Hubungan Faktor Resiko dengan Proses dan Produk
(sumber : software engineering risk management, dale walter karolak)

Untuk menerapkan model penyelesaian dari gambar 2 dan gambar 3, beberapa parameter dan persamaan harus dipertimbangkan. Dibawah ini beberapa persamaan yang digunakan untuk memecahkan masalah dari pohon kemungkinan.

1. $P(A) = [\sum_{n=1}^3 P(A_n)]/3$ asumsi bahwa setiap elemen resiko harus memiliki bobot nilai, jika bobot nilai setiap elemen berbeda, maka persamaan yang digunakan adalah $P(A) = w_1P(A_1) + w_2P(A_2) + w_3P(A_3)$ dimana w adalah angka positif dan $w_1 + w_2 + w_3 = 1$
2. $P(A_1) = [\sum_{n=4}^{13} w_nP(A_n)]$, dimana $w_4 = 0.64$, $w_5 = 0.64$, $w_6 = 0.64$, $w_7 = 0.71$, $w_8 = 0.71$, $w_9 = 0.55$, $w_{10} = 0.071$, $w_{11} = 0.71$, $w_{12} = 0.71$, $w_{13} = 0.71$. Berdasarkan tabel 1, maka bobot 0.55 merupakan nilai terendah, 0.64 nilai sedang dan 0.71 untuk nilai tinggi
3. $P(A_2) = [\sum_{n=4}^{13} w_nP(A_n)]$, dimana $w_4 = 0.66$, $w_5 = 0.66$, $w_6 = 0.66$, $w_7 = 0.66$, $w_8 = 0.50$, $w_9 = 0.66$, $w_{10} = 0.50$, $w_{11} = 0.50$, $w_{12} = 0.45$, $w_{13} = 0.66$. Berdasarkan tabel 1, maka bobot 0.45 merupakan nilai terendah, 0.50 nilai sedang dan 0.66 untuk nilai tinggi
4. $P(A_3) = [\sum_{n=4}^{13} w_nP(A_n)]$, dimana $w_4 = 0.62$, $w_5 = 0.62$, $w_6 = 0.62$, $w_7 = 0.55$, $w_8 = 0.45$, $w_9 = 0.55$, $w_{10} = 0.55$, $w_{11} = 0.45$, $w_{12} = 0.045$, $w_{13} = 0.62$. Berdasarkan tabel 1, maka bobot 0.45 merupakan nilai terendah, 0.55 nilai sedang dan 0.62 untuk nilai tinggi
5. $P(A_4) = [\sum_{n=1}^8 P(On)]/8$, dimana On adalah nilai matrik untuk nomor pertanyaan mengenai faktor resiko organisasi yang berjumlah 8 pertanyaan. Persamaan ini digunakan kembali untuk menyelesaikan matrik

resiko $P(A_5)$ sampai $P(A_{13})$ dengan mengubah nomor pertanyaan dari masing-masing faktor resiko.

6. $P(B) = \sum (Q1, Q2, Q3, Q4, Q5, Q9, Q10, Q11, Q12, Q14, Q15, Q16, Q17, Q18, Q19, Q21, Q22, Q23, Q24, Q28, Q30, Q35, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q45, Q46, Q47, Q48, Q49, Q60, Q77, Q78, Q79, Q80, Q81)/40$
7. $P(C) = \sum (Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q26, Q28, Q30, Q35, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q51, Q52, Q54, Q56, Q60, Q76)/34$
8. $P(D) = \sum (Q1, Q3, Q4, Q5, Q6, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q26, Q28, Q30, Q31, Q35, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q51, Q53, Q55, Q57, Q60, Q65, Q66, Q67, Q69)/38$
9. $P(E) = \sum (Q1, Q3, Q4, Q5, Q6, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q26, Q28, Q30, Q35, Q37, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q45, Q51, Q58, Q60, Q61, Q65, Q66, Q67, Q68, Q69, Q70)/39$
10. $P(F) = \sum (Q1, Q3, Q4, Q5, Q6, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q27, Q28, Q29, Q30, Q32, Q34, Q35, Q36, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q51, Q59, Q60, Q62, Q64, Q71, Q73, Q74, Q75, Q76)/42$
11. $P(G) = \sum (Q1, Q3, Q4, Q5, Q6, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q28, Q30, Q35, Q36, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q50, Q51, Q60, Q63, Q72)/33$
12. $P(H) = (\sum_{n=1}^{81} Q_n)/81$, dimana Q_n adalah seluruh pertanyaan yang ada pada matrik resiko
13. $P(I) = \sum (Q3, Q9, Q10, Q11, Q12, Q14, Q15, Q16, Q23, Q49, Q77)/12$
14. $P(J) = \sum (Q1, Q2, Q3, Q7, Q8, Q10, Q11, Q12, Q13, Q14, Q15, Q20, Q21, Q22, Q25, Q29, Q31, Q32, Q33, Q34, Q35, Q36, Q52, Q53, Q55, Q56, Q57, Q58, Q59, Q60, Q61, Q62, Q63, Q64, Q65, Q67, Q68, Q69, Q70, Q71, Q72, Q73, Q74, Q75, Q76, Q77)/46$
15. $P(K) = \sum (Q1, Q3, Q4, Q6, Q7, Q8, Q10, Q11, Q12, Q13, Q16, Q17, Q18, Q19, Q20, Q21, Q22, Q23, Q24, Q26, Q27, Q28, Q29, Q30, Q48, Q49, Q52, Q54, Q56, Q57, Q58, Q59, Q61, Q62, Q72, Q73, Q74, Q75, Q75, Q76, Q77, Q78, Q79, Q80)/42$
16. $P(L) = \sum (Q13, Q17, Q18, Q19, Q20, Q21, Q22)/7$
17. $P(M) = (\sum_{n=1}^{81} Q_n)/81$, dimana Q_n adalah seluruh pertanyaan yang ada pada matrik resiko.
18. $P(N) = [\sum_{n=4}^{13} w_n P(A_n)]$, dimana $w_4 = 0.55$, $w_5 = 0.55$, $w_6 = .0.65$, $w_7 = 0.75$, $w_8 = 0.65$, $w_9 = 0.72$, $w_{10} = 0.70$, $w_{11} = 0.72$, $w_{12} = 0.70$, $w_{13} = 0.60$.

Bobot 0.55 merupakan faktor resiko yang berpengaruh minor terhadap proses dan 0.72 merupakan faktor resiko yang berpengaruh mayor terhadap proses

$$19. P(O) = \left[\sum_{n=4}^{13} w_n P(A_n) \right], \text{dimana } w_4 = 0.69, w_5 = 0.45, w_6 = 0.55, w_7 = 0.50, w_8 = 0.40, w_9 = 0.65, w_{10} = 0.69, w_{11} = 0.69, w_{12} = 0.69, w_{13} = 0.69.$$

Bobot 0.45 merupakan faktor resiko yang berpengaruh minor terhadap produk dan bobot 0.69 merupakan faktor resiko yang berpengaruh mayor terhadap produk

6. HASIL PENELITIAN

Model SERIM yang terdapat pada tabel 4. memperlihatkan hasil penelitian dari pembangunan perangkat lunak akademik. $P(A)$ menjelaskan kesuksesan pembangunan perangkat lunak secara keseluruhan. $P(A_1)$ sampai $P(A_3)$ menunjukkan komponen elemen resiko. $P(A_4)$ sampai $P(A_{13})$ menunjukkan faktor resiko. $P(B)$ sampai $P(G)$ menjelaskan phase pembangunan life cycle dan $P(H)$ sampai $P(M)$ menjelaskan aktivitas resiko. Sedangkan $P(N)$ dan $P(O)$ menunjukkan resiko dari proses dan produk perangkat lunak.

Berdasarkan penelitian proyek perangkat lunak akademik memiliki nilai kemungkinan $P(A)$ 0.65, dimana nilai tersebut menyatakan bahwa proyek memiliki tingkat kesuksesan (keberhasilan) yang cukup baik. Untuk melihat kesuksesan suatu proyek, nilai $P(A)$ terletak antara nilai kemungkinan 0 (paling buruk) sampai 1 (sukses).

Tabel 4. juga memperlihatkan 3 (tiga) nilai kemungkinan terendah dari keberhasilan proyek sistem informasi akademik yaitu *tool* $P(A_8)=0.42$, *correctness* (P_{11})= 0.48 dan *reability* $P(A_{12})= 0.41$.

Rendahnya nilai pada faktor resiko *tool*, *correctness* dan *reability* memungkinkan pengambil keputusan untuk mengubah strategi untuk mengurangi atau menghindari resiko. misalnya proyek menerapkan *tool* yang dapat bekerja otomatis untuk pengujian perangkat lunak, sehingga jadwal yang ketat dalam penyelesaian proyek dapat diatasi.

Nilai kemungkinan terbaik terdapat pada *organization* $P(A_4)$ dan *personel* $P(A_{13})$ dengan nilai kemungkinan 0.71 dan 0.78, hal ini menerangkan bahwa unit UNIKOM center memiliki kemampuan sumber daya manusia dan pengalaman organisasi yang baik dalam membangun perangkat lunak akademik.

Tabel 4. Tabel SERIM penilaian perangkat lunak

Probabilitas Resiko	Nilai
P(A)	0.65
P(A ₁)	0.71
P(A ₂)	0.66
P(A ₃)	0.62
P(A ₄)	0.71
P(A ₅)	0.64
P(A ₆)	0.70
P(A ₇)	0.61
P(A ₈)	0.42
P(A ₉)	0.61
P(A ₁₀)	0.60
P(A ₁₁)	0.48
P(A ₁₂)	0.41
P(A ₁₃)	0.78

Tabel 4. Tabel SERIM penilaian perangkat lunak (lanjutan)

Probabilitas Resiko	Nilai
P(B)	0.63
P(C)	0.67
P(D)	0.56
P(E)	0.64
P(F)	0.73
P(G)	0.68
P(H)	0.69
P(I)	0.72
P(J)	0.66
P(K)	0.72
P(L)	0.66
P(M)	0.68
P(N)	0.72
P(O)	0.69

Berdasarkan tabel penilaian di atas, terlihat model SERIM dapat melakukan penilaian dan prediksi dari seluruh aspek resiko pembangunan perangkat lunak dengan dengan menyatukan seluruh phase pengembangan perangkat lunak, faktor resiko, elemen resiko, aktivitas resiko, proses dan produk. Model SERIM juga sangat membantu kita untuk memahami cara penggunaan model dalam manajemen resiko dengan menganalisis penilaian yang ada pada tabel 4.

7. KESIMPULAN

Paradigma manajemen resiko menggambarkan kumpulan aktivitas komunikasi secara berkelanjutan antara semua anggota proyek dan pelanggan. Biaya dan usaha dalam aktivitas dalam manajemen resiko harus lebih kecil dibanding keuntungan akan diperoleh perusahaan jika pengelolaan resiko proyek berhasil. Untuk mencapai keberhasilan tersebut, efisiensi manajemen resiko bisa meningkat dengan menggunakan alat bantu yang berbasis-komputer.

Sudah banyak alat bantu berupa program aplikasi yang dapat membantu untuk mengenali resiko pada proyek perangkat lunak seperti : *Risk Guide*, *Riskit* maupun *ProRisk*, tetapi aplikasi tersebut tidak dapat kita perbandingkan antara satu dengan yang lainnya, karena setiap pendekatan dari manajemen resiko dibangun didasarkan kondisi dan sudut pandang peneliti mengenai manajemen resiko, akibatnya berdampak pada cara peneliti dalam memecahkan masalah.

Perangkat lunak JIT merupakan salah satu model yang digunakan untuk mengelola resiko dalam proyek perangkat lunak dengan melihat resiko dari berbagai prespektif seperti : model proses, elemen resiko, faktor resiko dan aktivitas resiko. Kemampuan memprediksi resiko perangkat lunak diperlukan untuk mendukung kesuksesan membangun perangkat lunak. Hasil kesimpulan yang di dapatkan dari penelitian yang dilakukan adalah :

1. Penelitian di UNIKOM center menemukan beberapa faktor resiko yang menjadi perhatian para pengambil keputusan dalam menentukan tindakan

berupa pengurangan resiko pada proyek perangkat lunak akademik. Hasil penilaian pada matrik resiko menunjukkan rendahnya nilai pada faktor resiko *tool*, *correctness* dan *reability*, sehingga pengambil keputusan di UNIKOM center harus dapat menentukan tindakan berupa pengurangan resiko pada proyek. Salah satu bentuk pengurangan resiko tersebut adalah mengubah strategi yang telah ditetapkan semula dengan strategi yang baru, seperti : menerapkan *tool* yang dapat bekerja otomatis untuk pengujian perangkat lunak, sehingga jadwal yang ketat dalam penyelesaian proyek dapat diatasi.

2. Penerapan perangkat lunak JIT dapat mengenali hampir seluruh resiko proyek dan memberikan pengetahuan kepada pengembang perangkat lunak dalam mengelola, mengukur, menilai dan memprediksi resiko dengan menggunakan metodologi proses dan alat bantu yang dapat membuat setiap proyek perangkat lunak sukses dan berhasil.

Pendekatan JIT tidak membantu dalam memprediksi kualitas produk yang dihasilkan. Sangat diperlukan suatu pendekatan yang memungkinkan untuk memperkirakan resiko perangkat lunak dengan mengukur kehandalan produk yang dihasilkan secara rinci, oleh karena itu, seperti yang tersebut sebelumnya, di dalam penelitian ini, sengaja menghilangkan diskusi pendekatan tersebut dengan maksud membatasi lingkup dari penelitian.

Salah satu keterbatasan dari penelitian yang dilakukan adalah metode penerapan penyelesaian masalah dan objek penelitian yang masih tunggal. Perlu penelitian lebih lanjut untuk membandingkan perangkat lunak JIT dengan metode manajemen resiko lain, sehingga kehandalan dan validitas dari hasil yang diperoleh sesuai dengan apa yang diharapkan.

Keterangan :

- *) Peneliti adalah mahasiswa program doktor ilmu komputer Universitas Gadjah Mada, Yogyakarta

Daftar Pustaka

- [1] Boehm, B. W. *A Spiral Model of Software Development and Enhancement*, Computer. May, 61-72, 1988.
- [2] Boehm, B. W. *Software Risk Management: Principles and Practices*. IEEE Software, 8(1):32-41, 1991.
- [3] Crossman, Tevron D. *Software Quality in the fourth-Generation Technique Environment*. Data Processing 25, no 10, 2000
- [4] Grechenig, Thomas and James Zschernitz. *Making Code Metrics Useful for Practioners*. Proceedings of the third software Engineering Research Forum, Orlando, 1993
- [5] Lawler, RW. *System Perspective on Software Quality*. Proceedings of the fifth International Computer and Applications Conference, 2001

- [6] McClelland, S., *Organizational Needs Assessments : Design, Facilitation and Analysis*, Quorum Books, 1995
- [7] Karolak, D. 1998. *Software Engineering Risk Management*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1998.
- [8] Mc Call, J.A.P.K Richards and G.F. Walter. *Factors in Software Quality*. General Electric Command and Information System Tech. Report 77DIS02, Sunnyvale, 1977
- [9] Murine, Gerald E. *Applying Software Quality Metric*. Proceedings of the ASQC Congress transactions, Boston, 1983
- [10] Strategy of market software.2007. <http://www.idc.com>. diakses tanggal 12-10-2008

Aplikasi Helpdesk untuk Pencatatan Masalah dan Solusi Perbaikan Peralatan Komputer

Teddy Marcus Zakaria, Rina Angelina

Jurusan Teknik Informatika

Fakultas Teknologi Informasi Universitas Kristen Maranatha

Jl. Prof. drg. Suria Sumantri No. 65, Bandung 40164

Email : teddy.mz@maranatha.edu , aero_angel0@yahoo.com

Abstract

Right now, PT Akur Pratama has Yogya Dept. Store branches nearly everywhere. And there are numerous problems to handle in each of its branch such as software, hardware, and network. The constantly increasing number of branches owned made the burden in central office heavier which in turn made them unable to handle the problem accordingly. This means that center office could not verify whether the problem has been solved by Regional EDP or not. So, it makes the EDP unable to solve problem hierarchy.

With this help desk application, the increasing amount of problem could be handled hierarchy. Users could find the solution for a recurring problem via FAQ, which gathered solutions from numerous problems occurred in any branch. This application could also monitor RE performance in problem solving, and analyze its user so that less performing user could be informed. This application could also analyze problem recurrence probability, in which certain brands that is prone to problem could be marked so that the company could use more consideration before ordering.

Mainly, this application is expected to manage the problems that often occur in every branch of Yogya Dept. Store. So, the burden on the central office will be reduced. In addition, this application also increases the handling level of responsiveness to be preventive.

Keyword : software, hardware, network, Regional EDP, FAQ

1. Pendahuluan

Sekarang ini, PT Akur Pratama memiliki cabang toserba Yogya dimana-mana. Di setiap cabangnya terdapat berbagai masalah seperti *software*, *hardware*, dan *network*. Hal ini mengakibatkan:

- Beban di kantor pusat semakin berat.
- Masalah-masalah di setiap cabang kurang ditangani sesuai dengan hirarkinya.
- Kantor pusat tidak dapat memonitor apakah permasalahan setiap cabang toserba Yogya telah diselesaikan oleh Regional EDP atau tidak.
- Banyaknya masalah yang sama sering terjadi, membuat kantor pusat kerepotan dalam menanganinya.
- HM tidak mengetahui barang yang ingin diperbaiki dari cabang mana dan kapan barang tersebut diterima.

Hal tersebut mengakibatkan perusahaan ini membutuhkan suatu program yang dapat mengatasi masalah tersebut.

Tujuan aplikasi ini adalah agar perusahaan tersebut dapat mengatur masalah-masalah yang sering terjadi di setiap cabang toserba Yogya. Aplikasi ini juga bertujuan untuk menyediakan fasilitas *print* form untuk kategori *hardware* sebagai surat jalan dari barang yang ingin diperbaiki. Selain itu, aplikasi ini juga menaikkan level penanganan masalah dari responsif menjadi preventif.

Beberapa Daftar Singkatan yang digunakan pada tulisan ini.

Table 1 – Daftar Singkatan

Istilah	Singkatan dari	Keterangan
<i>EDP</i>	<i>Electronic Data Processing</i>	Staf disetiap cabang yang menangani bagian IT di cabang
<i>RE</i>	<i>Regional EDP</i>	Atasan EDP ditingkat regional
<i>EXPERT</i>	-	Teknisi IT yang khusus menangani masalah <i>hardware</i>
<i>NOA</i>	<i>Network & Office Automation</i>	Divisi IT yang khusus menangani masalah <i>network</i> .
<i>DMCAS</i>	<i>Data Management, Custom Application Support</i>	Divisi IT yang khusus menangani masalah <i>software</i> .
<i>HM</i>	<i>Hardware Management</i>	Divisi IT yang bertugas sebagai atasan <i>EXPERT</i> , yang khusus menangani masalah <i>hardware</i> .
<i>ADMIN</i>	<i>Administrator</i>	Atasan <i>DMCAS</i> , <i>NOA</i> dan <i>HM</i> ditingkat pusat
<i>GUI</i>	<i>Graphical User Interface</i>	Antar muka komputer yang berbasis grafis.
<i>SLA</i>	<i>Service Level Agreement</i>	Kesepakatan waktu penyelesaian masalah antara yang menangani masalah dengan <i>EDP</i> .

2. Teori / Kajian Pustaka

Helpdesk merupakan lapisan pertama yang harus dihubungi oleh *end user* bila mereka mendapatkan masalah. Helpdesk akan berupaya menanganinya, tapi bila gagal akan mengirimkan ke lapisan yang lebih senior. Selama itu, helpdesk akan menjadi koordinator dari penanganan masalah. *End user* harus selalu menghubungi helpdesk saat meminta bantuan ataupun menanyakan *progress* permintaan bantuan mereka. *End User* dilarang untuk menghubungi secara langsung lapisan support yang lebih dalam (mem-bypass helpdesk).

Berikut ini adalah penjelasan dari peran-peran yang umum ditemukan dalam menangani masalah *information technology*(IT) pada divisi:

- **Helpdesk**

Helpdesk adalah titik utama dimana *client* dari IT akan pertama kali menghubungi divisi IT saat mempunyai pertanyaan atau masalah yang berhubungan dengan IT. Helpdesk membawa harga diri dan wibawa divisi IT saat berhubungan dengan *client* sehingga Helpdesk sangat mempengaruhi *customer experience*.

Helpdesk menyimpan *database* dari masalah dan solusi yang muncul dari operasional IT sehari-hari. Helpdesk memfasilitasi komunikasi antara user dan bagian IT lainnya, merespon crisis, dan membuat prioritas pengerjaan masalah.

Karena merupakan titik pertama hubungan ke *client*, staf help desk harus mempunyai pengetahuan yang luas (meskipun tidak mendalam). Hal ini diperlukan agar sebuah masalah dapat segera dikategorikan dan diberikan pada tim solusi yang benar.

Helpdesk haruslah menjadi tempat utama *client* pertama kali menghubungi divisi IT. Bila tidak, penanganan masalah menjadi tidak terkoordinasi dan pengetahuan menjadi hilang setelah solusi diimplementasikan. *Client* tidak diperkenankan untuk menghubungi divisi lain karena akan mengacaukan prioritas kerja.

Helpdesk sebaiknya dibantu oleh *software* tertentu untuk memfasilitasi pelacakan sebuah insiden, eskalasi masalah, dan pelaporan. *Software* harus juga mampu melakukan pengkategorian masalah, menyimpan pengetahuan dari solusi yang didapat, dan melakukan prioritas pengerjaan.

- **End User Support**

End user support bertanggung jawab untuk perbaikan fisik komputer dan kunjungan ke lapangan kerja. Grup ini adalah lapisan kedua dari manajemen masalah dan solusi. Umumnya bila ukuran group cukup besar, manajer akan membagi menjadi beberapa tim kecil berdasarkan lokasi, teknologi, aplikasi, atau kelompok bisnis. Setiap kelompok kecil mempunyai seorang kepala.

Seperti Helpdesk, *End user support* harus juga mempunyai kemampuan yang luas pada sistem IT pada perusahaan. Perbedaannya, *End user support* mempunyai pengetahuan yang lebih mendalam pada sistem standar perusahaan. Keahlian lebih diarahkan pada *hardware* dan *software* yang ada pada sistem komputer *end user* bukan pada aplikasi server.

End user support bertanggung jawab dalam memberikan dukungan pada seluruh peralatan dan aplikasi yang terpasang pada sisi *end user*. Selain itu *End user support* juga bertanggung jawab pada instalasi peralatan baru, perawatan peralatan yang ada, dan upgrade pada sistem *end user*. Untuk memudahkan pekerjaan *End User Support*, IT Standard harus diberlakukan agar pekerjaan tidak terlalu beragam.

Selain kemampuan teknis, *end user support* harus mempunyai kemampuan untuk berkomunikasi dengan client dan membangun hubungan baik dengan anggota bisnis lain. Pekerjaan lainnya adalah memberikan training untuk *end user* sehingga mengurangi jumlah panggilan kepada *end user Support*.

Dalam sebuah organisasi IT yang lemah, adalah umum bila kita mendapati *end user / client* melompati helpdesk dan langsung menghubungi profesional atas. Bila terus berlangsung, sikap ini akan menimbulkan frustrasi pada profesional lapisan atas karena pekerjaan mereka yang terganggu. Ujung-ujungnya profesional atas akan keluar dari perusahaan saat moral kerja mereka menjadi terlalu rendah. Sifat dari *end user / client* ini juga menunjukkan frustrasi mereka pada IT karena merasa helpdesk kurang dapat membantu menangani masalah mereka.

- **Network Administration Group**

Network Administrator Group mengatur semua kemampuan jaringan komunikasi data yang dibutuhkan oleh bisnis. Network administrator bertanggung

jawab pada semua kabel, *hubs/switch*, keamanan jaringan, *routers*, *gateways*, *firewall*, dan hal yang berhubungan dengan jaringan lainnya. Mereka melakukan pengawasan *traffic* jaringan dan melakukan efisiensi / upgrade sebelum kebutuhan melebihi kapasitas.

Network administrator membutuhkan keahlian yang khusus meliputi pengetahuan pada hardware jaringan, *media network* / kabel, *network protocols*, enkripsi, dan *firewall*. Tingginya tuntutan keahlian dan pengetahuan pada network administrator menyebabkan tingginya pula pelatihan dan pengalaman yang harus dibayar agar seorang network administrator menjadi efektif. Pelatihan sendiri membutuhkan waktu 5 tahun lebih agar efektif.

Network administrator bertanggung jawab dalam meneliti aplikasi, akses, dan data transfer yang dibutuhkan. Kemudian menentukan solusi yang paling optimal dan menegosiasikan kontrak dengan vendor. Penilaian kebutuhan, perencanaan kapasitas, dan implementasi yang baik dapat mengurangi biaya.

Untuk perusahaan menengah atau kecil, network administrator dan system administrator dapat dikerjakan oleh satu orang.

3. Desain Perangkat Lunak

3.1. Identifikasi Kebutuhan Sistem

Dalam tahap ini akan dijelaskan mengenai identifikasi aplikasi program yang akan dibuat. Dilakukan dengan cara penelitian dan konsultasi dengan pekerja pada PT Akur Pratama. Di dalam tahap identifikasi ini diperoleh *informasi* yaitu:

- Aplikasi yang dibutuhkan adalah aplikasi untuk mengatur masalah-masalah yang sering terjadi di setiap cabang toserba Yogya.
- Aplikasi harus dapat mencatat nama cabang yang mengirim barang dan tanggal terima barang di HM.
- Aplikasi ini dapat melihat jumlah pertanyaan yang telah dijawab dan jumlah pertanyaan yang belum dijawab.
- Aplikasi ini dapat menghitung rata-rata perbaikan.
- Aplikasi ini dapat mengatur pertanyaan EDP dibidang *software* atau *network* langsung ke RE untuk menjawabnya. Jika RE tidak mampu menjawabnya, RE harus memberikan alasan untuk melakukan eskalasi ke DMCAS atau NOA.
- Aplikasi ini dapat membuat report per cabang, per merk, per sub kategori.

3.2. Overview Sistem

Sistem yang akan dikembangkan untuk PT Akur Pratama setelah dilakukan identifikasi kebutuhan adalah sebagai berikut:

- Aplikasi yang dibutuhkan adalah sistem pengaturan jawaban pertanyaan berdasarkan hirarkinya.
- Aplikasi ini mencatat nama cabang yang mengirim barang dan tanggal terima barang pada menu jawaban HM.
- Aplikasi ini dapat menghitung jumlah pertanyaan yang telah dijawab dan jumlah pertanyaan yang belum dijawab berdasarkan status pertanyaan.
- Aplikasi ini dapat menghitung rata-rata perbaikan *hardware* berdasarkan total waktu penyelesaian dibagi jumlah pertanyaannya.
- Aplikasi ini dapat mengatur pertanyaan EDP dengan melihat kategori dan status pertanyaannya

- Aplikasi ini dapat menampilkan report per cabang, per merk dan per sub kategori berdasarkan data yang telah diperoleh dari *database*.

3.3. Desain Perangkat Lunak Secara Keseluruhan

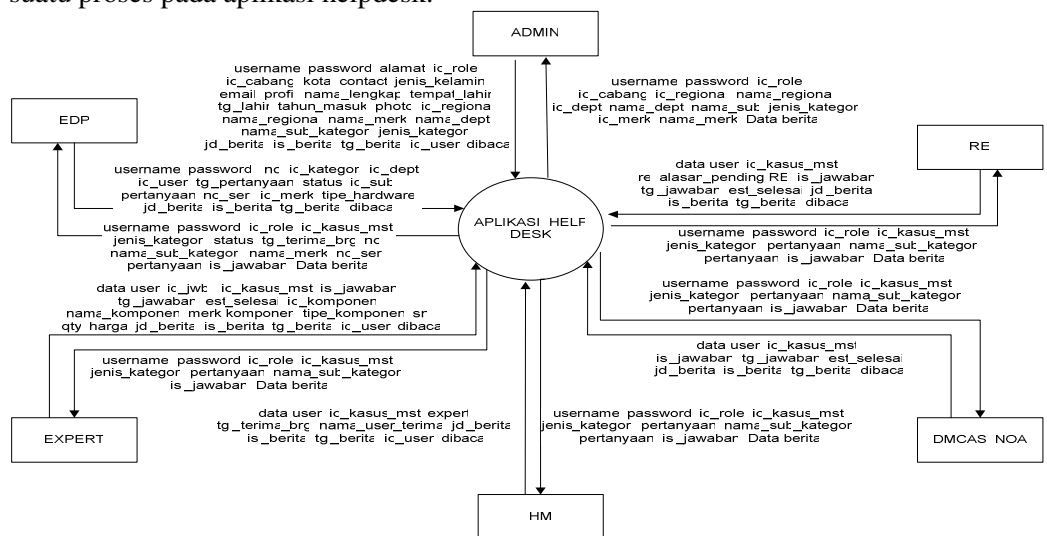
Berikut ini merupakan gambaran dari aplikasi helpdesk dalam bentuk *Data Flow Diagram* (DFD) dan perancangan *database* melalui *Entity Relationship Diagram* (ERD) yang kemudian diterjemahkan dalam bentuk *Entity Relationship Table*.

Data Flow Diagram (DFD)

Proses yang terjadi di dalam aplikasi digambarkan melalui data flow diagram. Setiap proses yang terjadi ditulis data *input* dan data outputnya di setiap level.

DFD LEVEL 0

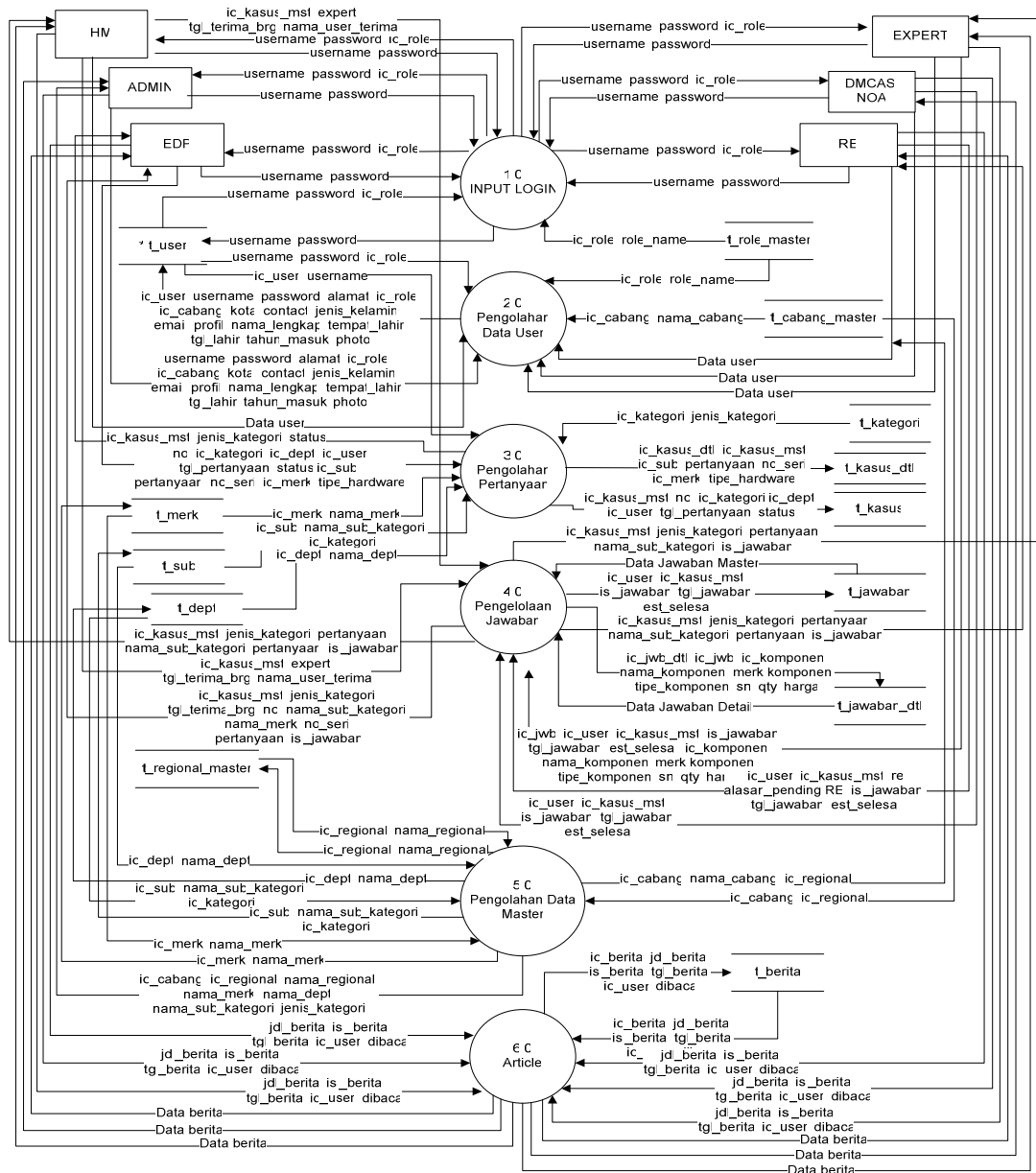
Gambar 1 menggambarkan dfd level 0 atau biasa disebut *context diagram*. Pada proses ini EDP, RE, DMCAS, NOA, EXPERT, HM dan ADMIN melakukan suatu proses pada aplikasi helpdesk.



Gambar 1 DFD Level 0

DFD LEVEL 1

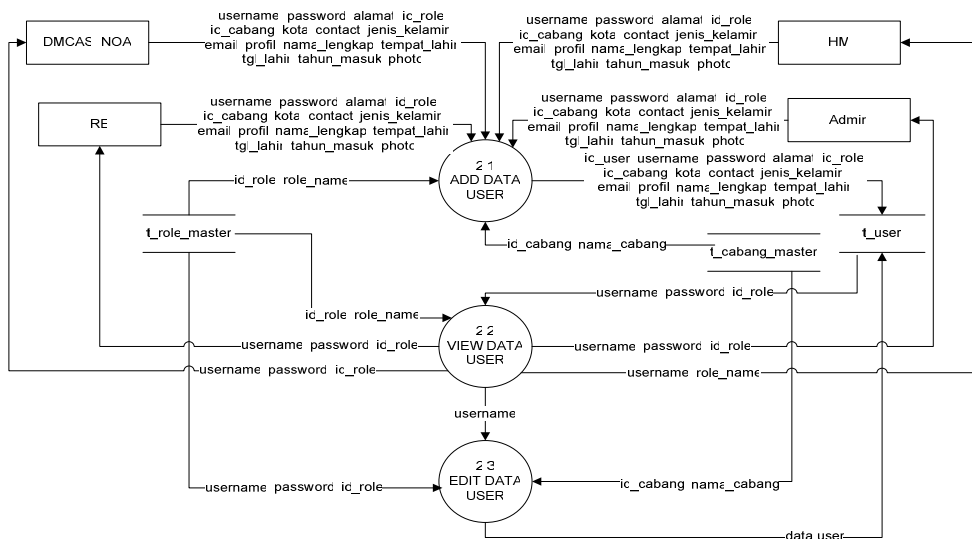
Pada gambar 2 dapat terlihat keseluruhan proses yang terjadi apabila menjalankan aplikasi Helpdesk. Proses yang terjadi adalah proses *input* login, proses pengelolaan data *user*, proses pengolahan pertanyaan, proses pengolahan jawaban, proses pengolahan data master, proses article.



Gambar 2 DFD Level 1

DFD LEVEL 2 dari Proses 2

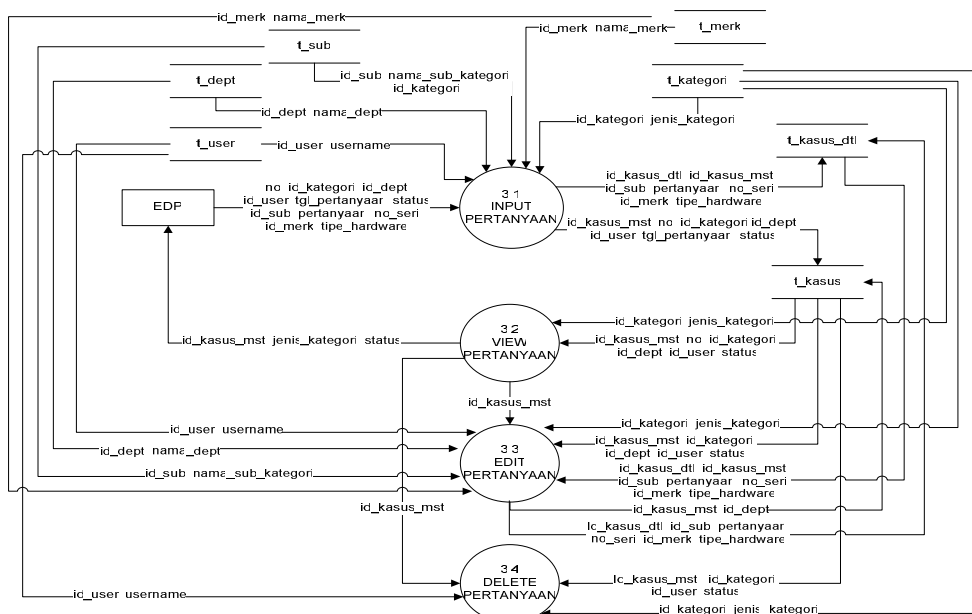
Gambar 3 merupakan proses DFD level 2 dari proses pengolahan data *user*. Pada proses ini *add*, *view*, dan *edit* data *user* tergantung dari *id_role* yang melakukan login. RE hanya dapat melakukan *add*, *view*, dan *edit* data *user* EDP. DMCAS dan NOA dapat melakukan *add*, *view*, dan *edit* data *user* EDP dan RE. HM dapat melakukan *add*, *view*, dan *edit* data *user* EDP, RE, dan EXPERT. ADMIN dapat melakukan *add*, *view*, dan *edit* data *user* EDP, RE, EXPERT, DMCAS, NOA dan HM.



Gambar III.3 DFD Level 2 Proses 2

DFD LEVEL 2 dari Proses 3

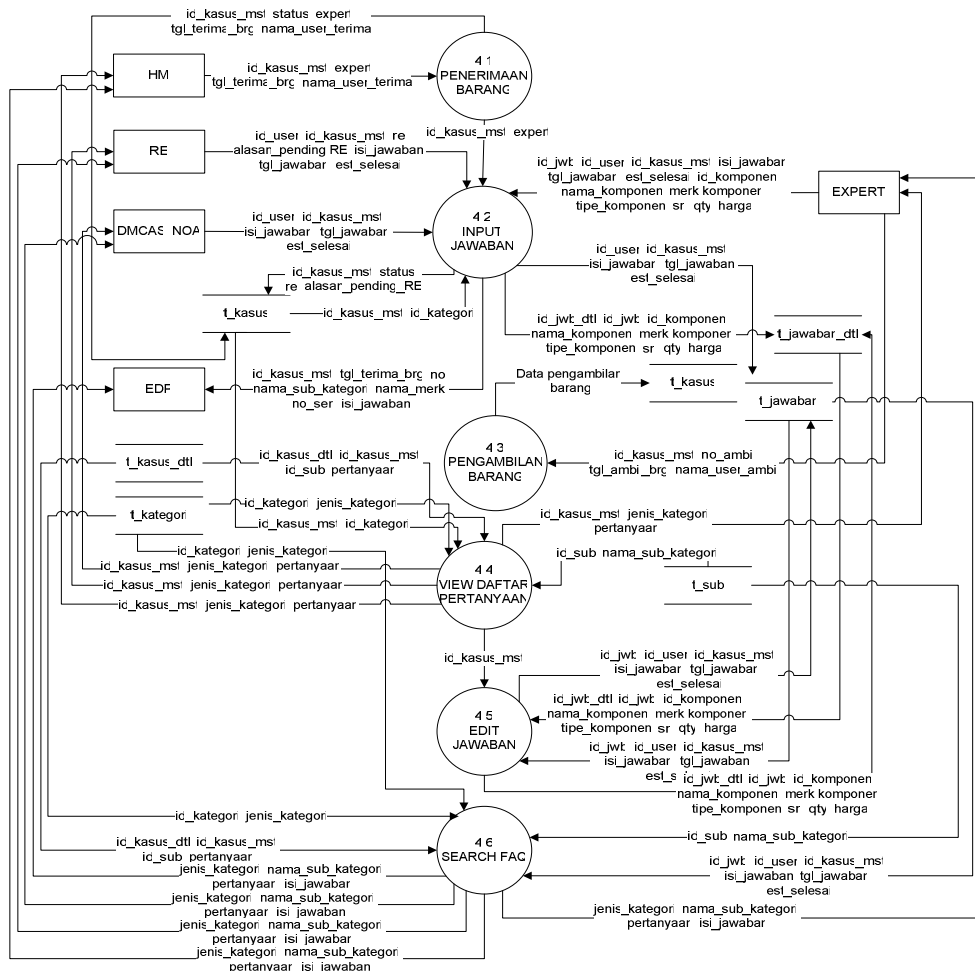
Gambar 4 merupakan proses DFD level 2 dari proses pengolahan pertanyaan. Pada proses ini hanya EDP yang dapat melakukan proses yang ada didalamnya. Proses yang terdapat pada gambar tersebut adalah proses *input* pertanyaan, *view* pertanyaan, *edit* pertanyaan dan *delete* pertanyaan.



Gambar 4 DFD Level 2 Proses 3

DFD LEVEL 2 dari Proses 4

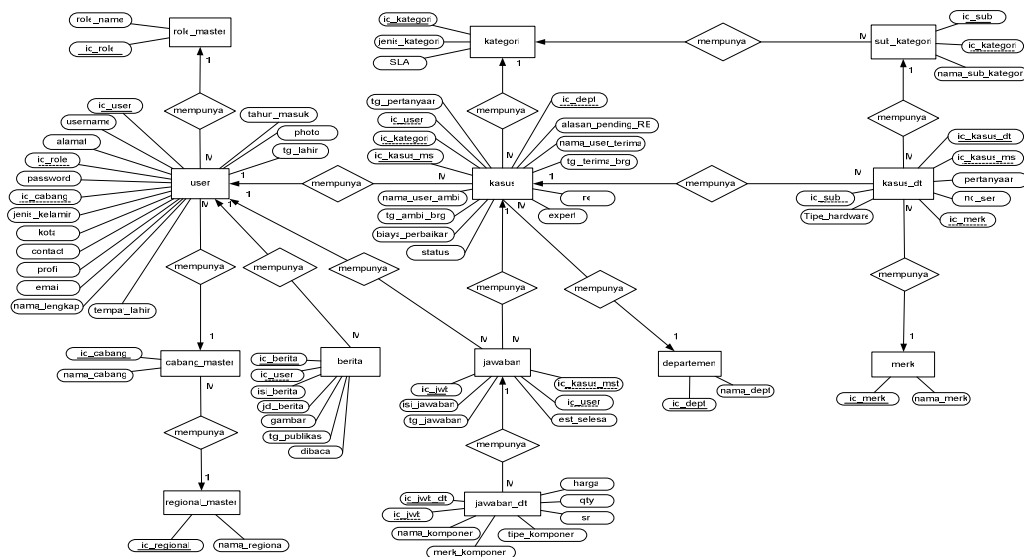
Gambar 5 merupakan proses DFD level 2 dari proses pengolahan jawaban. Pada proses ini yang dapat melakukan proses yang ada didalamnya. Proses yang terdapat pada gambar tersebut adalah proses *input* pertanyaan, *view* pertanyaan, *edit* pertanyaan dan *delete* pertanyaan.



Gambar 5 DFD Level 2 Proses 4

DFD LEVEL 2 dari Proses 5

Gambar 6 merupakan proses DFD level 2 dari proses pengolahan data master. Pada proses ini hanya dapat dilakukan oleh ADMIN. Pada level ini proses yang terjadi adalah proses *input*, *view* dan *edit* untuk data cabang, regional, departemen, sub kategori dan merk.

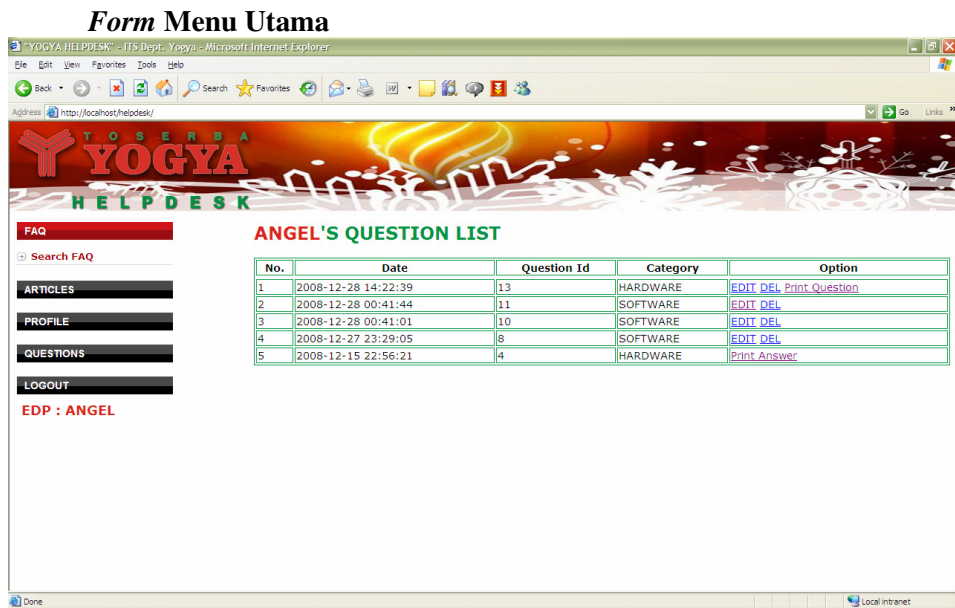


Gambar 7 ERD Helpdesk

4. Implementasi

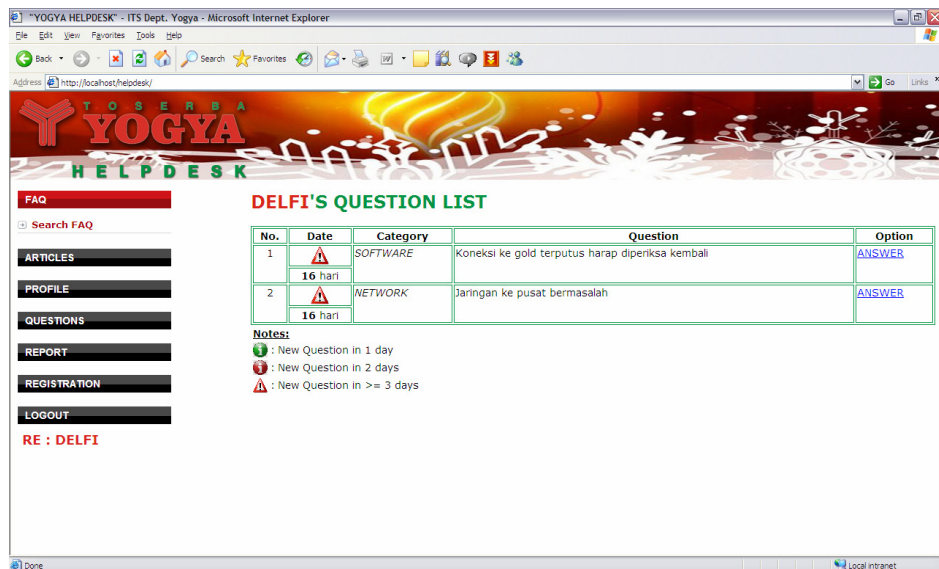
Implementasi dari desain perangkat lunak ini memiliki fitur-fitur berikut :

- **Fitur Login**
Fitur ini bertujuan agar *user* dapat mengakses menu utama aplikasi ini sesuai dengan hak aksesnya.
- **Fitur New Question**
Ini adalah fitur untuk membuat pertanyaan baru dalam kategori *software*, *network* dan *hardware*. Fitur ini hanya dapat dilakukan oleh EDP.
- **Fitur Answer Question**
Fitur ini bertujuan untuk menjawab pertanyaan baru dalam kategori *software*, *network* dan *hardware*. Fitur ini hanya dapat dilakukan oleh RE, DMCAS dan NOA untuk kategori *software* dan *network*, serta EXPERT untuk kategori *hardware*. Jika RE tidak dapat menjawab pertanyaan dari EDP, RE dapat memberikan alasan untuk melakukan eskalasi kepada DMCAS atau NOA.
- **Fitur Search FAQ**
Fitur ini bertujuan untuk melihat pertanyaan-pertanyaan yang telah selesai dijawab. Fitur ini dapat dilakukan oleh semua *user*. Pada fitur ini kategori yang ditampilkan hanya *software* dan *network*.
- **Fitur Article**
Ini adalah fitur untuk memasukkan artikel seputar teknologi informasi. Fitur ini dapat dilakukan oleh semua *user*. Pada fitur ini *user* dapat melihat artikel yang dibuat oleh *user* lainnya.



Gambar 8 Form Menu Utama EDP

Untuk form menu utama EDP, di dalamnya berisi menu search FAQ, article list, new article, view profile, change *password*, question list, new question dan logout.



Gambar 9 Form Menu Utama RE

Form menu utama RE berisi menu search FAQ, article list, new article, view profile, change *password*, question list, new *user*, *user* list dan logout.

- **Form New Question**

The screenshot shows a web browser window titled "YOGYA HELPDESK - ITS Dept. Yogya - Microsoft Internet Explorer". The address bar shows "http://localhost/helpdesk/". The page has a header with the "YOGYA HELPDESK" logo and a navigation menu on the left with links: FAQ, ARTICLES, PROFILE, QUESTIONS (highlighted), and LOGOUT. Under "QUESTIONS", there are links for "Question List" and "New Question". The "New Question" form is displayed with the following fields: User Id (1), Full Name (RINA ANGELINA), Contact (08130445678), Category (SOFTWARE), Department (Departemen Mechanical & Electrical), and Branch Name (GRIYA ANTAPANE). There are "Insert Question" and "BACK" buttons at the bottom of the form. The user's name "EDP : ANGEL" is shown at the bottom left of the page.

Gambar 10 Form New Question

Gambar ini merupakan form New Question, dimana *user* harus memilih *category* dan *department*. Jika kategori yang dimasukkan adalah *software* atau *network* maka *user* hanya dapat menginput 1 pertanyaan per 1 id_kasus_mst. Jika kategori yang dimasukkan adalah *hardware* maka *user* dapat memasukkan lebih dari 1 pertanyaan per 1 id_kasus_mst .

- **Form Answer**

The screenshot shows a web browser window titled "YOGYA HELPDESK - ITS Dept. Yogya - Microsoft Internet Explorer". The address bar shows "http://localhost/helpdesk/". The page has a header with the "YOGYA HELPDESK" logo and a navigation menu on the left with links: FAQ, ARTICLES, PROFILE, QUESTIONS (highlighted), and LOGOUT. Under "QUESTIONS", there are links for "Question List" and "New Question". The "Answer Question" form is displayed with the following fields: Date of Question (2009-12-28 00:41:44), User Id (1), Full Name (RINA ANGELINA), Contact (08130445678), Category (SOFTWARE|YGHOSM), Department (Departemen Mechanical & Electrical), and Branch Name (GRIYA ANTAPANE). There is a "Question" field with a text area containing "ada s
". The "Status" field is set to "ANSWER|A". There are "Date of Answer" (2009-01-08) and "Finish Est." (2009-01-08) fields. There is a "Answer" field with a text area. There are "SUBMIT" and "BACK" buttons at the bottom of the form. The user's name "RE : DELFI" is shown at the bottom left of the page.

Gambar 11 Form Answer RE

Gambar ini merupakan form Answer Question RE, dimana *user* dapat memilih status. Jika *user* memilih statusnya answer, maka *user* harus mengisi date

of answer, finish est., dan answer .Jika *user* memilih statusnya escalation, maka *user* harus memasukkan alasan eskalasi. Lalu tekan tombol Submit.

The screenshot shows the 'YOGYA HELPDESK' web application. The left sidebar contains navigation links: FAQ, ARTICLES, PROFILE, QUESTIONS (selected), REPORT, REGISTRATION, LOGOUT, and DMCAS : SUSIE. The main content area is titled 'ANSWER QUESTION'. It contains the following fields:

- Date of Question : 2008-12-27 23:29:05
- User Id : 1
- Full Name : RINA ANGELINA
- Contact : 08180445678
- Category : SOFTWARE | YGHOSM
- Question : aaddc88
- Escalation Reason : aad
- Date of Answer : 2009-01-08
- Finish Est. : 2009-01-08
- Answer : (text area)

Buttons: SUBMIT, BACK

Gambar 12 *Form Answer DMCAS atau NOA*

Gambar ini merupakan form Answer Question DMCAS atau NOA, dimana *user* harus mengisi date of answer, finish est., dan answer . Lalu tekan tombol Submit.

The screenshot shows the 'YOGYA HELPDESK' web application. The left sidebar contains navigation links: FAQ, ARTICLES, PROFILE, QUESTIONS (selected), REPORT, REGISTRATION, LOGOUT, and HM : AEP. The main content area is titled 'ESCALATION QUESTION'. It contains the following fields:

- Date of Question : 2008-12-28 14:22:39
- User Id : 1
- Full Name : RINA ANGELINA
- Contact : 08180445678
- Sub Category : ADAPTOR
- Brand : ASUS
- Hardware Type : sa12345
- Serial Number : 123
- Question : sad
- Received Date : 2009-01-08
- Received By : (text area)
- Expert : ERVINA

Buttons: SUBMIT, BACK

Gambar 13 *Form Answer HM*

Gambar ini merupakan form Answer HM yang berupa eskalasi pertanyaan, dimana *user* harus mengisi received date, received by, dan memilih expert yang akan menjawabnya. Lalu tekan tombol Submit.

YOGYA HELPDESK

ANSWER EXPERT'S QUESTION

Date of Question : 2009-12-15 23:13:00

User Id : [] Department : Departemen Mechanical & Electrical

Full Name : RINA ANGELINA Branch Name : GRIYA ANTAPANI

Contact : 08190445678

Receipt Date : 2009-12-28

No	Sub Category	Brand	Hardware Type	Serial Number	Question
1	HANDPHONE	HP	sa342	12345	tidak mau hidup
2	CD-RW	ASUS	sa345	123212	rusak

Insert Component

No	Component Id	Component Name	S/N	Qty	Price	Option
Summary Price : Rp. []						

Finish Est. : 2009-01-08

Answer : []

SUBMIT BACK

Gambar 14 Form Answer EXPERT

Gambar ini merupakan form Answer EXPERT yang telah dieskalasi oleh HM, dimana *user* harus data komponen dengan menekan tombol Insert Component. Kemudian Summary Price akan menjumlah total harga komponen yang dimasukkan. Setelah itu, *user* memasukkan Finish est., dan answer. Lalu tekan tombol Submit.

- **Form Report**

YOGYA HELPDESK

QUESTION'S REPORT

Date : 2009-01-01 until 2009-02-16

Branch : GRIYA ANTAPANI

Category : *** CHOOSE ***

Sub Category : *** All ***

Status : All Questions

SUBMIT

Notes:

- New Question in 1 day
- New Question in 2 days
- New Question in >= 3 days

Gambar 15 Form Report

Pada gambar ini, *user* dapat memilih *date*, *branch*, *category*, *sub category*, status. Lalu tekan Submit untuk menemukan data yang dicari.

4. Kesimpulan dan Saran

Aplikasi Helpdesk dapat mengatur permasalahan yang sering terjadi di setiap cabang toserba Yogya, sehingga beban di kantor pusat akan berkurang. Aplikasi ini menyediakan fasilitas print pada form perbaikan, form selesai service, dan form transfer barang sebagai surat jalan dari barang yang ingin diperbaiki. Selain itu, aplikasi ini juga menaikkan level dari responsif menjadi preventif.

Sistem berjalan dengan baik, semua fungsi yang dibuat dapat berjalan dengan baik dan sesuai dengan yang diharapkan. Dan alur penjawaban pertanyaan sudah sesuai dengan yang diinginkan. Sehingga hasil yang didapat mendekati akurat untuk dilapangan.

Saran yang dapat diberikan atas aplikasi ini apabila akan dikembangkan di kemudian hari adalah adanya hubungan antara aplikasi helpdesk dengan aplikasi PPBJ. Supaya tabel jawaban detail dapat dihubungkan ke tabel komponen yang ada pada aplikasi PPBJ.

DAFTAR PUSTAKA

- [1]. Arsitektur Perangkat Lunak, http://id.wikipedia.org/wiki/Arsitektur_perangkat_lunak, 19/05/2008
- [2]. Help Desk, http://en.wikipedia.org/wiki/Help_desk, 16/01/2009
- [3]. Hartono, Jogyanto. (1999). Analisis dan Desain Sistem Informasi. Andi. Yogyakarta.
- [4]. Pandarion. (2008). Struktur Organisasi IT dan Peran Divisi IT. Available: <http://pandarion.wordpress.com/2008/11/09/struktur-organisasi-it-dan-peran-divisi-it/>. Accessed: 10/02/2009.
- [5]. Siswoutomo, Wiwit. (2005). PHP Enterprise Kiat Jitu Membangun Web Skala Besar. PT Elex Media Komputindo. Jakarta.
- [6]. Suteja, Bernard Renaldy, Agus Prijono, Rusdy Agustaf. (2005). Mudah dan Cepat Menguasai Pemrograman Web. Informatika. Bandung.

Sistem Informasi Training & Development di HRD – PT. X

Radiant Victor Imbar, Evlin Marcelline Fendrianto

Jurusan Sistem Informasi

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri no. 65 Bandung 40164

Email: radiant.vi@eng.maranatha.edu , evlin.marcelline@student-partners.com

Abstract

The improvement in information technology has been increased day by days along with the more requests to make the business process in the company will be much efficient and secure. In company, the need of information system is very important now days. PT X is a company that has more than 1000 staff so they need a system to manage human resources start from the recruitment process, personnel administration, payroll, training and development, personnel cost planning. One of the human resources management that must be maintained is training and development process. This paper will explain about the implementation of training and development process that build an software that run on web based application and has several functions such as create training, preparation of resources for training, booking training, and reporting.

Keywords: Training and Development, Human Resources, Software.

1. Latar Belakang

PT. X terdiri dari PT. XITbk. F&Y Division, PT. XITbk. P&Y Division, PT SPD, SYD, PT NW W&Y Division, PT. XITbk. PFD, PT. XITbk. N&M Division, dan Bank JA. PT. XFI didukung oleh ISD (Information System Division) dimana di dalamnya didukung oleh beberapa modul yaitu *Network & Basis*, MM Module (Material Management), PP Module (Production Planning), PM Module (Plant Maintenance), SD Module (Sales & Distribution), FICO Module (Finance & Control), ABAP dan HR Module (Human Resource). Hal – hal yang ditangani oleh HR Module (Human Resource) ini yaitu mengenai training & development (T&D), kesehatan, personalia, transportasi, dan satpam, dimana T&D ini mencakup juga research & development (R&D), recruitment, dan training. Bagian training di T&D ini menangani semua proses training yang terjadi di PT. X dari mulai perencanaan sampai menghasilkan laporan mengenai training tersebut, dimana di dalamnya terdapat pencatatan perencanaan modul training, peserta yang ikut dalam training, waktu dari training tersebut akan dijalankan, dan sosialisasi mengenai training tersebut yang berupa catalog. Pada saat ini, divisi – divisi yang ada diminta untuk migrasi sistem operasi dari Windows ke Linux. Bagian T&D ini menggunakan aplikasi HRPuzzle yang berjalan di sistem operasi Windows dan tidak dapat diimplementasikan pada sistem operasi Linux. Oleh karena itu, sebagai solusinya akan dibuatkan migrasi system HRPuzzle ke system WEB sehingga dapat diakses di sistem operasi Linux.

2. Tujuan Pembuatan Sistem

Tujuan pembuatan system ini adalah untuk mengetahui proses bisnis T&D yang belum terkomputerisasi, menganalisa dan melakukan migrasi aplikasi HRPuzzle ke Web sehingga bisa diakses di sistem operasi LINUX.

3. Pembatasan Masalah

Batasan masalah akan dibagi menjadi 3 yaitu :

3.1 Perangkat Lunak

- Sistem operasi : Microsoft Windows XP Professional SP 2
- Sistem Basis Data : Oracle
- Bahasa Scripting : PHP, XHTML, Java script, AJAX
- Editor Pemograman : Macromedia Dreamweaver 8.0, PHP Designer 2007
- Web Server : Php Triad

3.2 Perangkat Keras

Server untuk pembuatan aplikasi

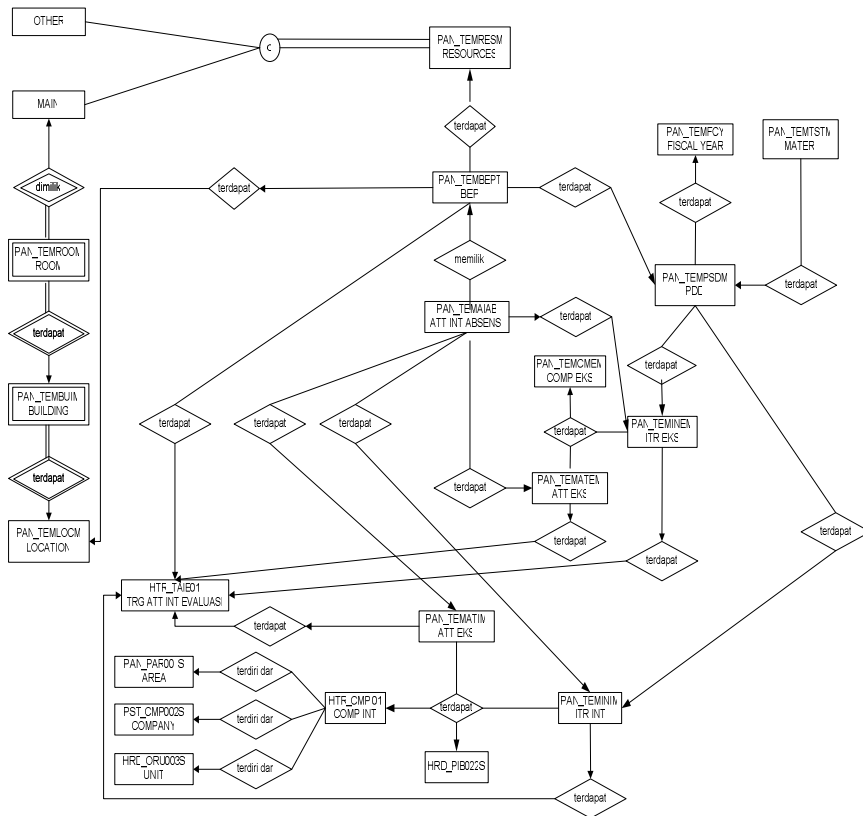
- *Processor Intel Pentium III 736 MHz*
- *Memory SDR 256 Mb.*
- *Harddisk 40 Gb*
- *Keyboard + Mouse*

3.3 Aplikasi

- Aplikasi ini digunakan pada bagian T&D HRD PT.X.
- Aplikasi ini dibagi menjadi 7 hak akses: *director, manager, section chief, transportasi, auditor, personalia dan seksi.*
- Aplikasi ini tidak menangani proses pembayaran.
- Keamanan diberikan pada bagian login (*encrypt password*) dan pembatasan menu dengan (*role*).

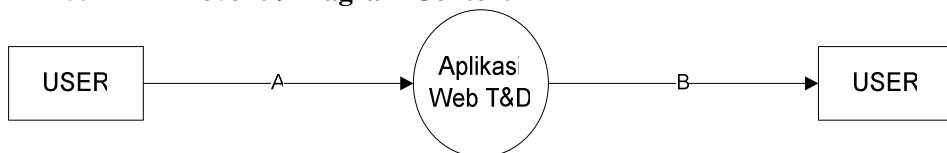
4 Entity Relationship Diagram

Berikut ini adalah gambar ER diagram :



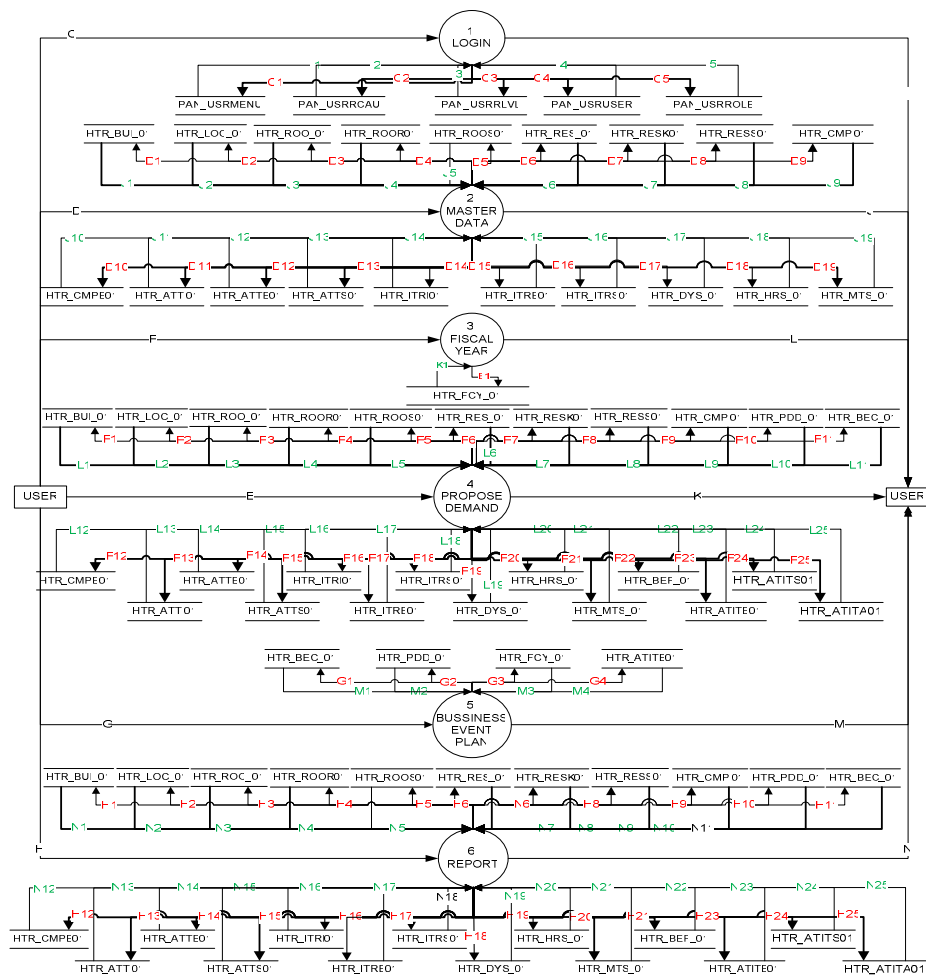
5 DFD (Data Flow Diagram)

5.1 DFD Level 0 / Diagram Context



FD Level 0 dari aplikasi web T&D ini menjelaskan mengenai proses utama, dimana *user* meminta layanan kepada sistem dan sistem memberikan balikan kepada pengguna.

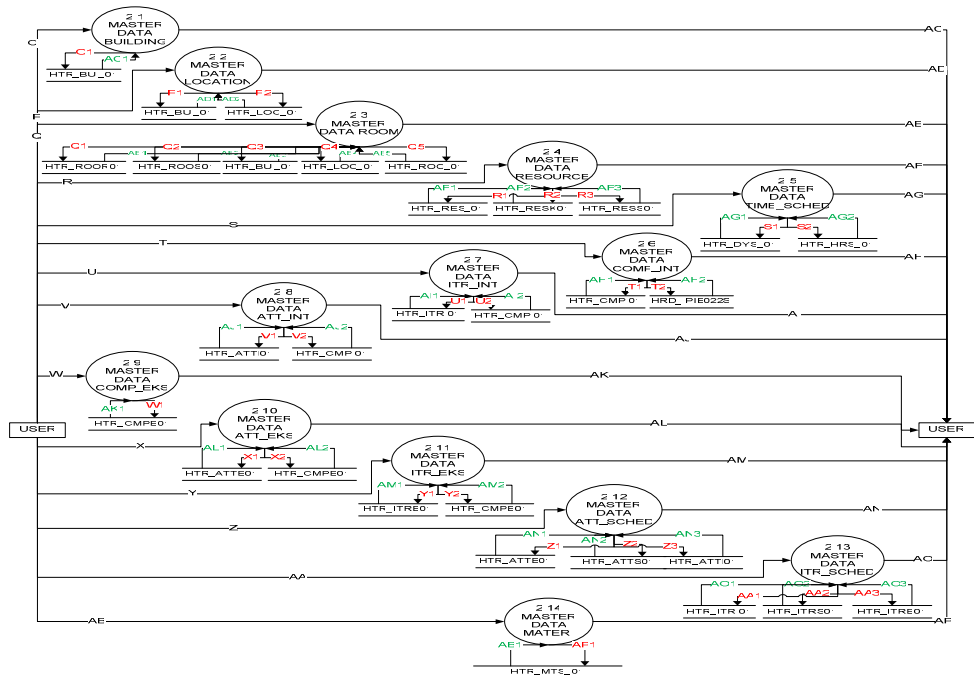
5.2 DFD Level 1 / Diagram Context



DFD Level 1 ini menjelaskan 6 besaran pada sistem web ini yaitu *login* (proses utama dari web yang memberikan layanan keamanan dan hak akses), *master data* (layanan data utama yang menjadi syarat awal pengajuan pelatihan), *fiscal year* (batasan tahun untuk proses pelatihan yang berlangsung), *propose demand* (proses proposal pelatihan), *business event plan* (proposal yang telah disetujui yang kemudian menjadi bahan acuan untuk evaluasi dan penilaian) dan *report* (laporan yang dihasilkan atas pelatihan yang telah diajukan baik telah terlaksana atau belum terlaksana).

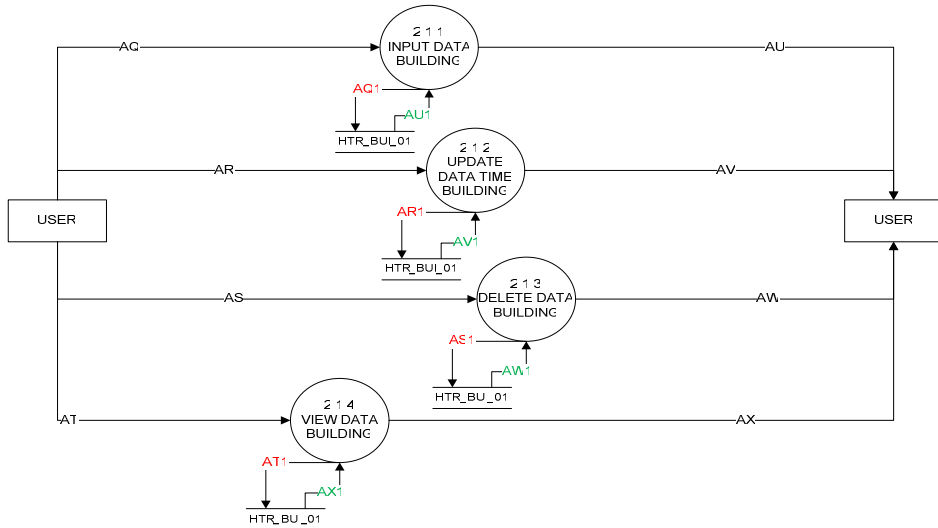
5.3 DFD LEVEL 2 PROSES 1

5.4 DFD *LEVEL* 2 PROSES 2



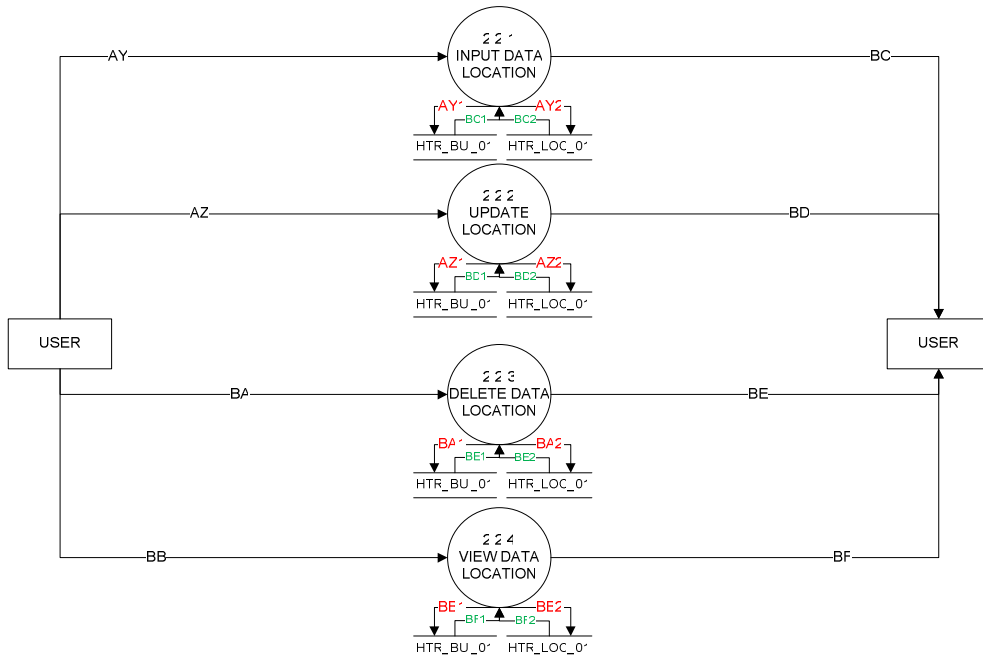
95

5.5 DFD LEVEL 3 PROSES 2.1



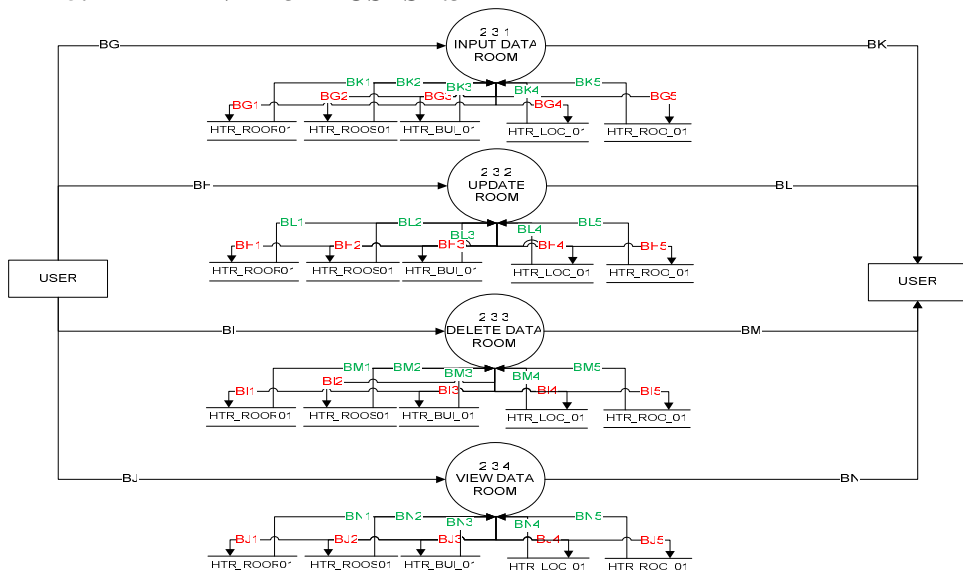
DFD Level 3 Proses 2.1 ini menjelaskan layanan master data *building*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.6 DFD LEVEL 3 PROSES 2.2



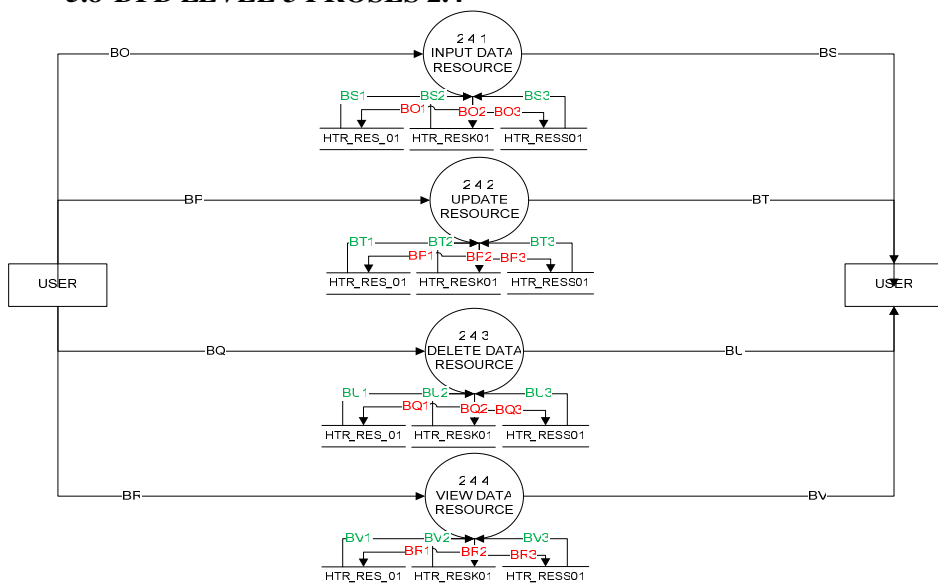
DFD Level 3 Proses 2.2 ini menjelaskan layanan master data *location*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.7 DFD LEVEL 3 PROSES 2.3



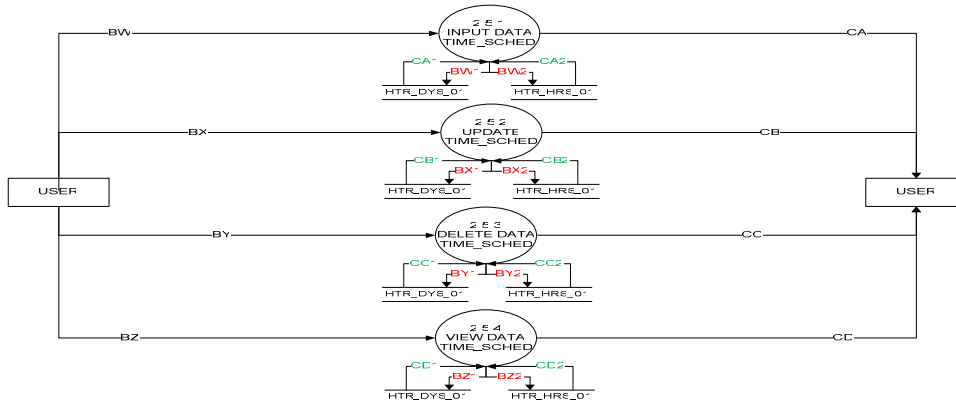
DFD Level 3 Proses 2.3 ini menjelaskan layanan master data *room*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.8 DFD LEVEL 3 PROSES 2.4



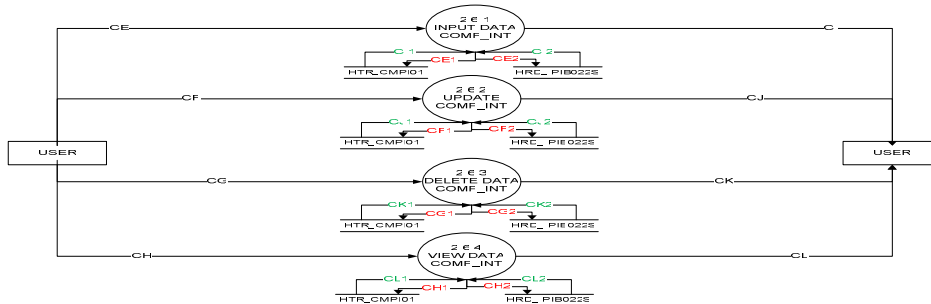
DFD Level 3 Proses 2.4 ini menjelaskan layanan master data *resource*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.9 DFD LEVEL 3 PROSES 2.5



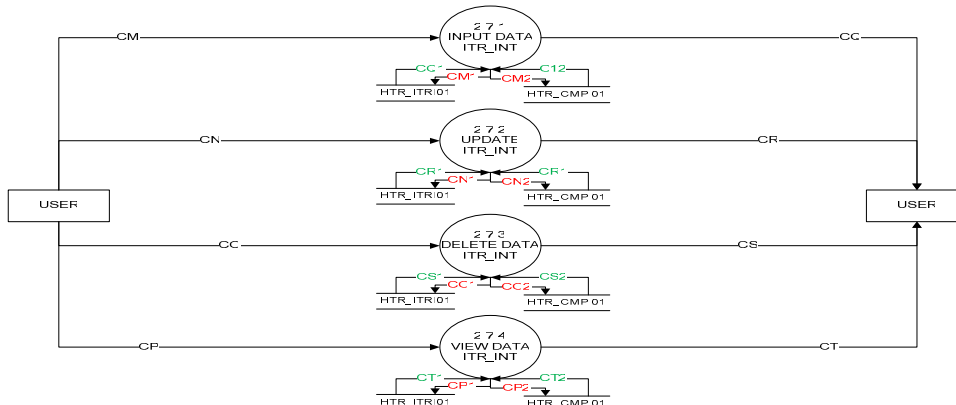
DFD Level 3 Proses 2.5 ini menjelaskan layanan master data *time schedule*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.10 DFD LEVEL 3 PROSES 2.6



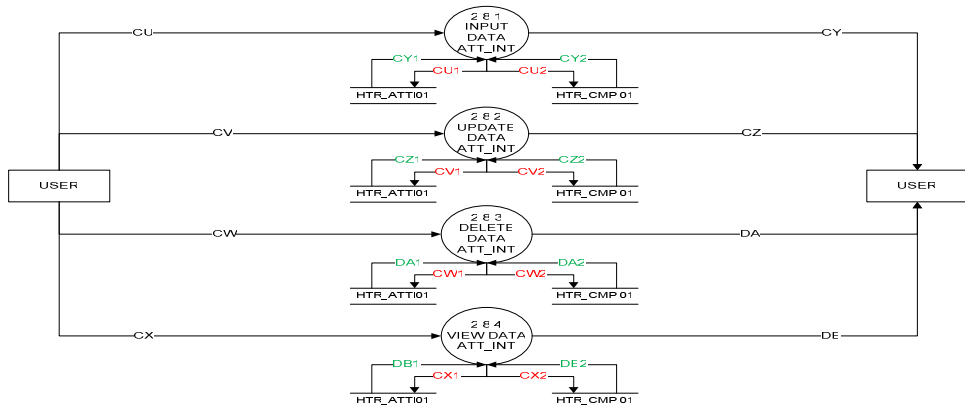
DFD Level 3 Proses 2.6 ini menjelaskan layanan master data *company internal*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.11 DFD LEVEL 3 PROSES 2.7



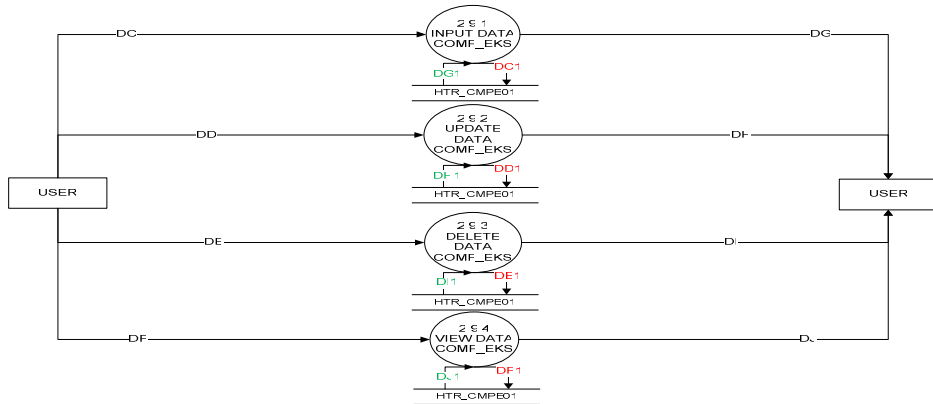
DFD Level 3 Proses 2.7 ini menjelaskan layanan master data *instruktur internal*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.12 DFD LEVEL 3 PROSES 2.8



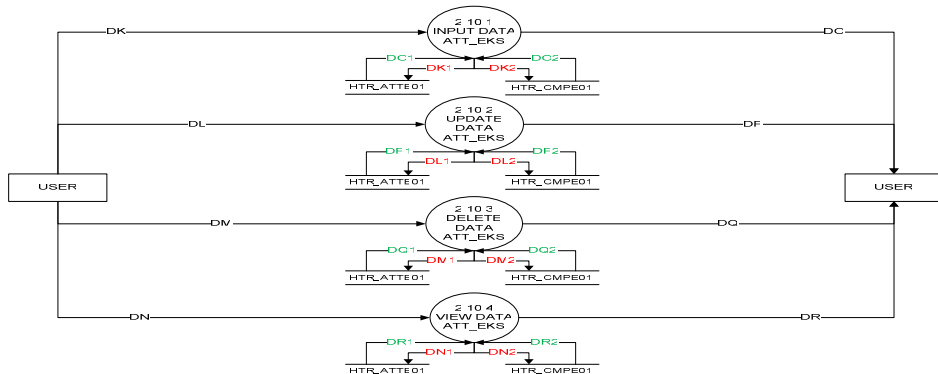
DFD Level 3 Proses 2.8 ini menjelaskan layanan master data *attendee internal*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.13 DFD LEVEL 3 PROSES 2.9



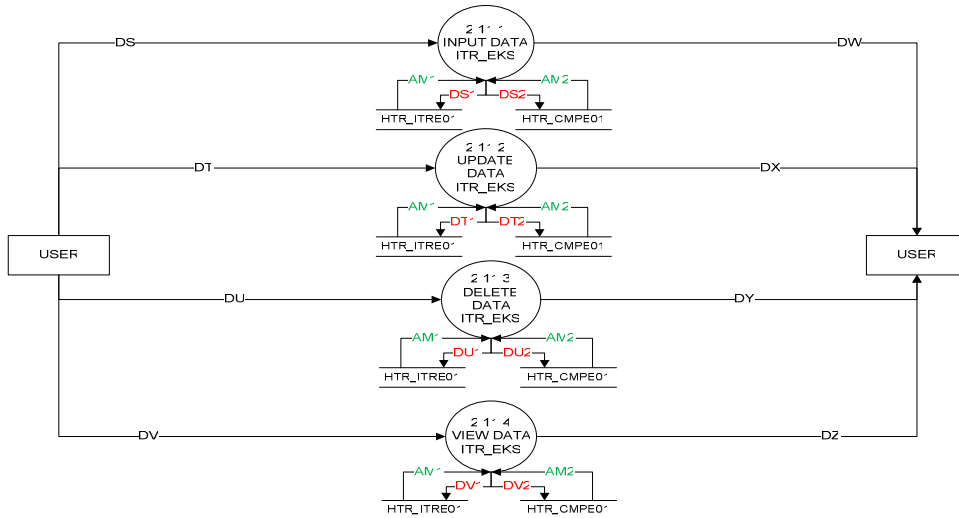
DFD Level 3 Proses 2.9 ini menjelaskan layanan master data *company eksternal*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.14 DFD LEVEL 3 PROSES 2.10



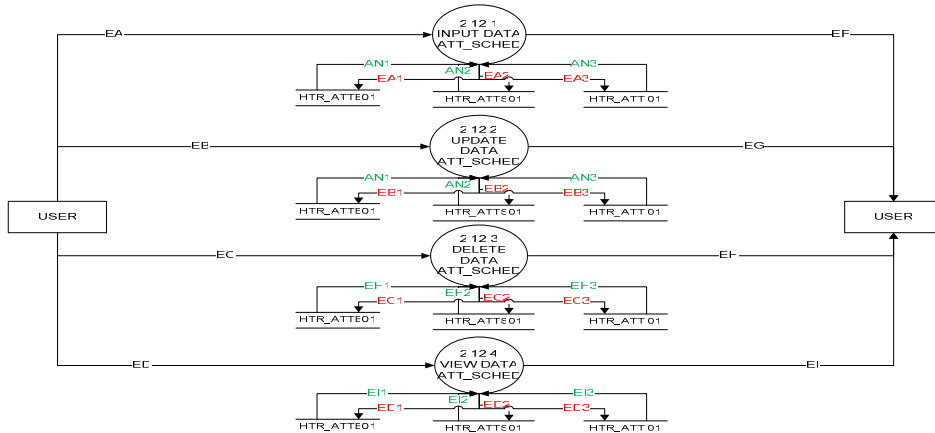
DFD Level 3 Proses 2.10 ini menjelaskan layanan master data *attendee eksternal*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.15 DFD LEVEL 3 PROSES 2.11



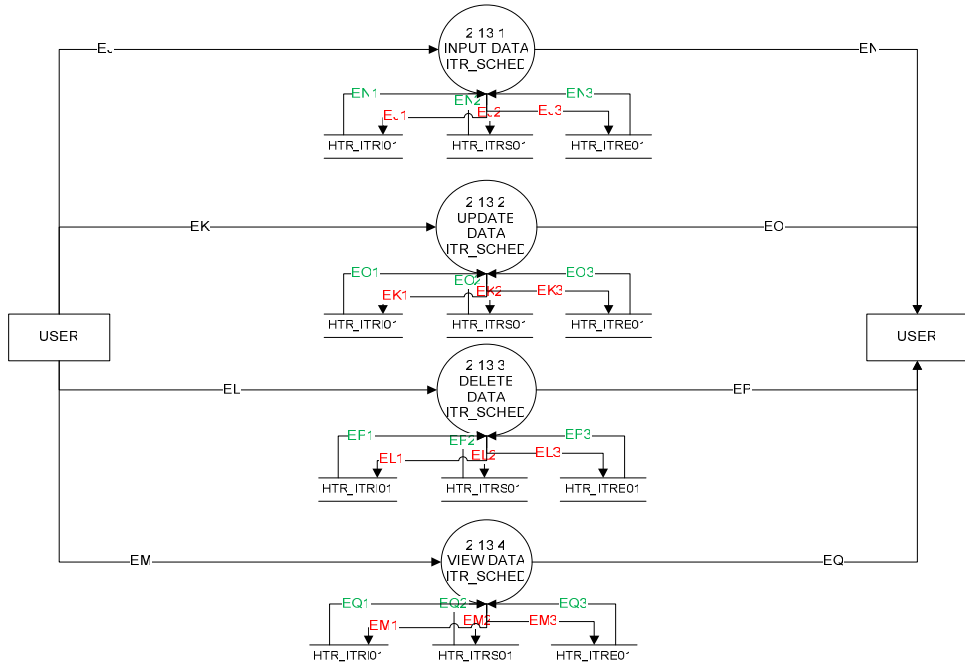
DFD Level 3 Proses 2.11 ini menjelaskan layanan master data *instructor internal*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.16 DFD LEVEL 3 PROSES 2.12



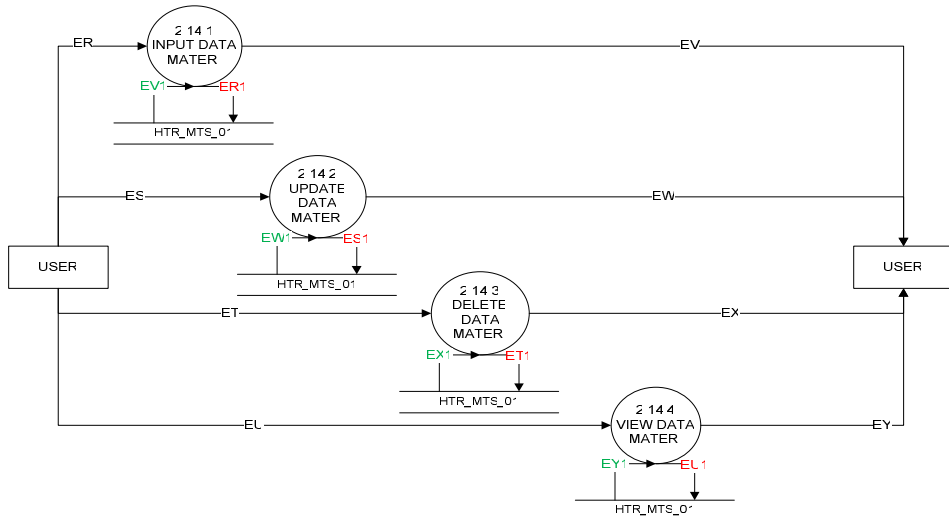
DFD Level 3 Proses 2.12 ini menjelaskan layanan master data *attendee schedule*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.17 DFD LEVEL 3 PROSES 2.13



DFD Level 3 Proses 2.13 ini menjelaskan layanan master data *instructor schedule*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

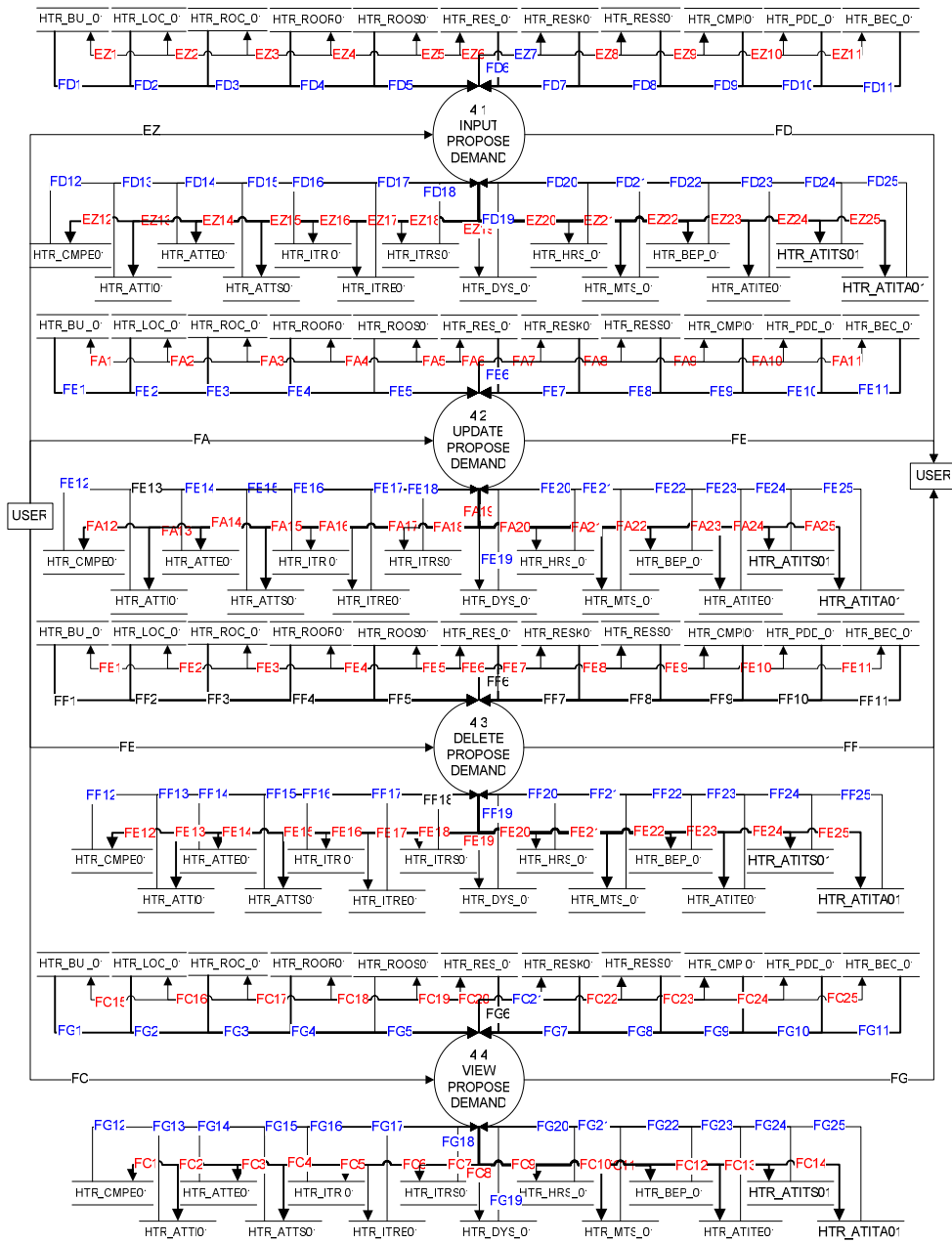
5.18 DFD LEVEL 3 PROSES 2.14



FD Level 3 Proses 2.14 ini menjelaskan layanan master data *materi*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

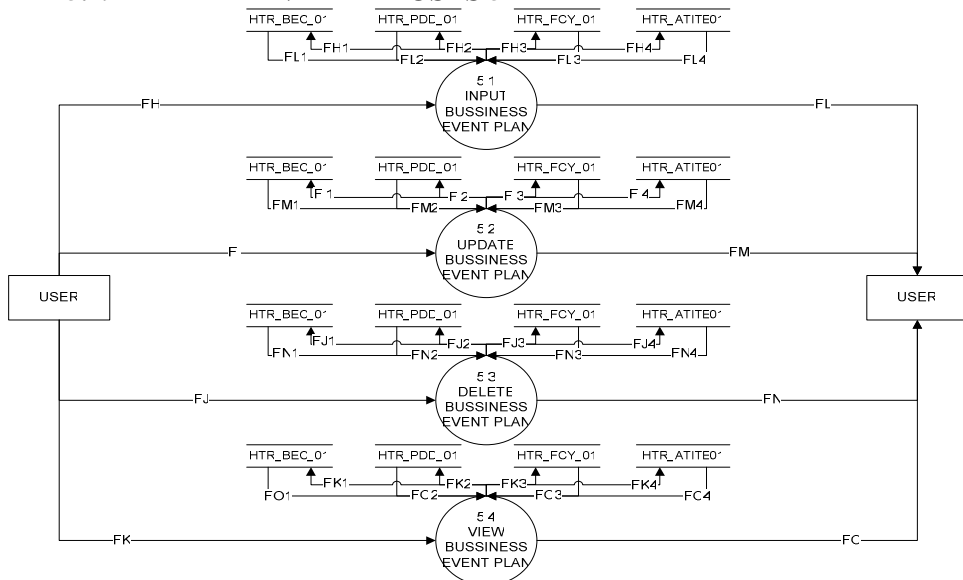
D

5.19 DFD LEVEL 2 PROSES 4



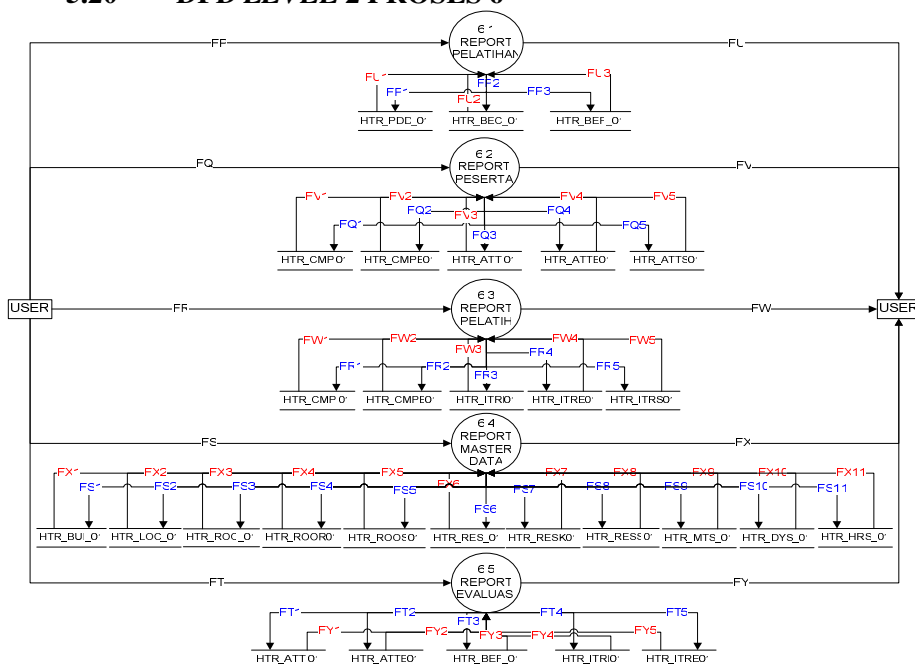
DFD Level 2 Proses 4 ini menjelaskan layanan *propose demand*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.19 DFD *LEVEL 2* PROSES 5



DFD Level 2 Proses 5 ini menjelaskan layanan *business event plan*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

5.20 DFD *LEVEL 2* PROSES 6



DFD Level 2 Proses 6 ini menjelaskan layanan master data *attendee schedule*, jadi pengguna dapat tambah, ubah, hapus dan lihat data.

6 Kamus Data

Berikut ini adalah contoh 2 buah kamus data yang dibuat :

6.1 Kamus Data *Building*

Nama_Data	Data_Building
Deskripsi	Data mengenai <i>Building</i>
Struktur Data	<p>Data_Building = @id_pbui + source + category + short_name + long_name + street_city + province + nation + postal_code + telephone + fax + contact_person + email_address + weblink + keterangan + created_date + created_by + modify_date + modify_by</p> <p>id_pbui = 3{A - Z}+3{A - Z}4{MIC XIV I}+1{0!...!9}</p> <p>source = {A - Z la - z}</p> <p>category = {A - Z la - z}</p> <p>short_name = {A - Z la - z}</p> <p>long_name = {A - Z la - z}</p> <p>street_city = {A - Z la - z}</p> <p>province = {A - Z la - z}</p> <p>nation = {A - Z la - z}</p> <p>postal_code = {A - Z la - z}</p> <p>telephone = {A - Z la - z}</p> <p>fax = {A - Z la - z 0 - 9}</p> <p>contact_person = {A - Z la - z}</p> <p>email_address = {A - Z la - z 0 - 9 @ .. }</p> <p>weblink = {A - Z la - z 0 - 9 @ .. }</p> <p>keterangan = {A - Z la - z}</p> <p>created_date = {A - Z la - z}</p> <p>created_by = {A - Z la - z}</p> <p>modify_date = {A - Z la - z}</p> <p>modify_by= {A - Z la - z}</p>

6.2 Kamus Data Propose Demand

Nama_Data	Data_Propose_Demand
Deskripsi	Data mengenai pengajuan pelatihan
Struktur Data	<p>Data_Propose_Demand = @id_ppdd + no + periode + company + seksitpk + subtpk + peserta_jpo + peserta_jgl + peserta_jv + tgl_rencana + nik_itrnama_itrjabatan_itr + status + created_date + created_by + modify_date + modify_by</p> <p>id_ppdd = {A - Z la - z 0 - 9}</p> <p>no = {0 - 9}</p> <p>periode = {A - Z la - z 0 - 9}</p> <p>company = {A - Z la - z 0 - 9}</p> <p>seksitpk = {A - Z la - z 0 - 9}</p> <p>subtpk = {A - Z la - z 0 - 9}</p> <p>peserta_jpo = {0 - 9}</p> <p>peserta_jgl = {0 - 9}</p> <p>peserta_jv = {0 - 9}</p> <p>tgl_rencana = {A - Z la - z 0 - 9}</p> <p>nik_itr = {A - Z la - z 0 - 9}</p> <p>nama_itr = {A - Z la - z 0 - 9}</p> <p>jabatan_itr = {A - Z la - z 0 - 9}</p> <p>status = {A - Z la - z 0 - 9}</p> <p>created_date = {A - Z la - z}</p> <p>created_by = {A - Z la - z}</p> <p>modify_date = {A - Z la - z}</p> <p>modify_by= {A - Z la - z}</p>

7 PSPEC(PROSES SPESIFIKASI)

Berikut ini 3 buah contoh PSPEC yang dibuat :

7.1 PSPEC data Login

No.Proses	1.1
Nama Proses	<i>Input Data Login</i>
Deskripsi	Proses digunakan untuk login <i>user</i> sebelum menggunakan aplikasi
<i>Input</i>	Data Login (<i>username</i> , <i>password</i>)
<i>Output</i>	Data Menu, Data Role, Data <i>User</i> , Data Company, Data <i>Level</i>
Nama Prosedur	Button_login
Logika Proses	<ol style="list-style-type: none"> 1. <i>user</i> memasukkan data login (<i>username</i>, <i>password</i>) 2. sistem memeriksa ke <i>database</i> dan tabel data login yang dimasukkan 3. sistem mengembalikan pesan status benar / salah data yang dimasukkan 4. jika salah, maka akan tampil pesan kesalahan berupa message box dan <i>user</i> harus mengisi ulang data login 5. jika benar, maka akan tampil ke tampilan berikutnya

7.2 PSPEC data user

No.Proses	1.2
Nama Proses	Verifikasi data <i>User</i>
Deskripsi	Proses digunakan untuk verifikasi data <i>user</i> terhadap role dan <i>level</i> yang dia miliki beserta hak akses menunya
<i>Input</i>	Data Company (<i>company</i> , <i>area</i>)
<i>Output</i>	Data <i>User</i> , Data <i>Level</i> , Data Menu
Nama Prosedur	Button_verifikasi
Logika Proses	<ol style="list-style-type: none"> 1. <i>user</i> memasukkan data company (<i>company</i> , <i>area</i>) 2. sistem memeriksa data login yang sebelumnya telah dimasukkan dengan memeriksa data company yang dimasukkan ke dalam <i>database</i> dan tabel 3. sistem mengembalikan pesan status role menu yang dimasukkan 4. sistem akan menampilkan form_id yang sesuai dengan hak login dari <i>user</i> 5. jika ada kesalahan pengisian company, dapat diperbaiki pada menu change companny

7.3 PSPEC Master Data Building

No.Proses	2.1
Nama Proses	Master Data <i>Building</i>
Deskripsi	Proses digunakan untuk tambah baru, hapus, ubah dan lihat data <i>building</i>
<i>Input</i>	Data <i>Building</i> (<i>id_pbui</i> , <i>source</i> , <i>category</i> , <i>short_name</i> , <i>long_name</i> , <i>street_city</i> , <i>province</i> , <i>nation</i> , <i>postal_code</i> , <i>telephone</i> , <i>fax</i> , <i>contact_person</i> , <i>email_address</i> , <i>weblink</i> , keterangan , <i>created_date</i> , <i>created_by</i> , <i>modify_date</i> , <i>modify_by</i>)
<i>Output</i>	Info <i>Building</i>
Nama Prosedur	Button_verifikasi

Logika Proses

1. *user* memilih menu *building* (menu ini akan tampil atau tidak sesuai dengan hak aksesnya)
2. memasukkan data *building* (*id_pbui* , *source* , *category* , *short_name* , *long_name* , *street_city* , *province* , *nation* , *postal_code* , *telephone* , *fax* , *contact_person* , *email_address* , *weblink* , *keterangan* , *created_date* , *created_by* , *modify_date* , *modify_by*)
3. sistem memeriksa data *building* yang dimasukkan
4. sistem mengembalikan pesan status dari data yang dimasukkan
5. jika data sudah benar, sistem akan menyimpan data ke *database*
6. jika data masih salah, akan tampil pesan kesalahan berupa message box dan kemudian *user* harus membenarkan data yang salah dimasukkan

8 Implementasi Program

Berikut ini beberapa contoh *screen shot* program :

8.1 Halaman Utama

The screenshot shows the 'Master Data Room' interface. On the left is a sidebar with navigation links: Personal Development, Recruitment Information Management, Training & Event Management, Master Data, Fiscal Year Business, Propose Demand, Business Event Plan, Attendance Report, Business Event Report, and Resource Report. The main area is titled 'Master Data Room' and has a breadcrumb trail: Personal Development > Training & Event Management > Edit Room. It contains a search bar with 'SINGKATAN' selected, a 'Cari Building' button, and a table of building data. Below the table are 'Print Preview' and 'Download' buttons.

checkbox	ID LOCATION	ID BUILDING	SHORT NAME	LONG NAME	TELEPHONE	FAX	CONTACT PERSON	EMAIL ADDRESS	WEBLINK	KETERANGAN
<input checked="" type="checkbox"/>	LOC_I_0001	BUI_I_0001	VC	VINSEN COMPUTER	08122092233	84467377	-	vin_2233@yahoo.com	http://www.vin-comp.com	-
<input checked="" type="checkbox"/>	LOC_I_0001	BUI_I_0002	TM	TOKO MARESHA	081931160037	91516706	-	shasha_0911@yahoo.com	http://www.shs.com	-
<input type="checkbox"/>	LOC_I_0002	BUI_I_0003	ST	SUGANTO SHOP	0	0	-	nama@alamat.eks	http://	-
<input type="checkbox"/>	LOC_I_0002	BUI_I_0004	US	USEP SHOP	0	0	-	nama@alamat.eks	http://	-
<input type="checkbox"/>	LOC_I_0002	BUI_I_0005	TS	TATANG SHOP	0	0	-	nama@alamat.eks	http://	-
<input type="checkbox"/>	LOC_I_0002	BUI_I_0006	VS	VENTI SHOP	0	0	-	nama@alamat.eks	http://	-
<input type="checkbox"/>	LOC_I_0002	BUI_I_0007	OS	OPK SHOP	0	0	-	nama@alamat.eks	http://	-

Tampilan di atas ini tampil setelah pengguna memilih menu personal, menu yang tampil pada sebelah kiri layar pengguna, akan berbeda-beda setiap pengguna karena disesuaikan dengan hak aksesnya yang telah diatur dalam *role* oleh *administrator*.

8.2 Display Master Data Location

The screenshot shows the 'Master Data Location' interface. On the left is a sidebar with navigation links: Training & Event Management, Master Data, Location, Building, Room, Resources, Topik Sub-Topik, Company, Instructor Internal, Instructor Eksternal, Business Event Catalog, Display Location, Display Building, Display Room, and Display Resources. The main area is titled 'Master Data Location' and has a breadcrumb trail: Personal Development > Training & Event Management > Edit Location. It contains a search bar with 'SOURCE' selected, a 'Cari Location' button, and a table of location data. Below the table are 'Print Preview' and 'Download' buttons.

checkbox	ID LOCATION	SOURCE	CATEGORY	SHORT NAME	LONG NAME	STATUS	ADDRESS	CITY	PROVINCE	NATION	POSTAL_CODE	TELEPHONE	FAX	CONTACT_PERSON	EMAIL_ADD
<input type="checkbox"/>	LOC_E_0001	EKSTERNAL	HOTEL	LM	LENMARC	AKTIF	KOPO MAS	BANDUNG	JAWA BARAT	IND	40225	undified	5430422	FANNY	gracia@met
<input type="checkbox"/>	LOC_I_0001	INTERNAL	COMPANY	DCL	DECALL29	AKTIF	SURYA SUMANTRY	BANDUNG	JAWA BARAT	INDONESIA	40123	12345	54322	EVJIN	a@a.a

8.3 Baru Propose Demand

Personnel Development

Formulir Analisis Kebutuhan Pelatihan (FAKP)

Personal Development > Training & Event Management > Tambah FAKP

Baru Ubah/Hapus Tambah ITR Tampil

Cari Berdasarkan TOPIK

Nilai

Cari Topik

Daftar TOPIK & SUBTOPIK berdasarkan :
Ike %%

SEQ	ID_TPK	TOPIK	ID_STPK	SUBTOPIK
04	2	SAP	4	ENTITY RELATION DIAGRAM
05	2	SAP	5	CUSTOMER RELATIONSHIP
06	3	MLTM	6	VIDEO
07	3	MLTM	7	AUDIO
02	1	SR	3	WORK UTILITIES
09	3	MLTM	9	DIMENSI
01	1	SR	1	WORK INSTRUCTION
02	1	SR	4	ENTITY RELATION DIAGRAM
08	3	MLTM	8	FLASH

FORMULIR ANALISIS KEBUTUHAN PELATIHAN

NO

ID_PROPOSE DEMAND

ID_TOPIK

TOPIK

ID_SUBTOPIK

SUBTOPIK

UNIT / PERUSAHAAN

ID_SEKSI

SEKSI / DEPARTEMEN

RENCANA PELAKSANAAN KEGIATAN PELATIHAN

FISCAL YEAR

START PERIODE

END PERIODE

Tampilan di atas ini digunakan untuk tambah data *propose demand*. Untuk menampilkan ini, pengguna harus login terlebih dahulu, kemudian pilih *company* dan *area* yang sesuai dengan hak aksesnya, kemudian pilih menu *personnel development* → *propose demand* → *fakp*. Fungsi – fungsi yang disediakan untuk memudahkan pengguna adalah fungsi pencarian (dimana pengguna cukup memilih kriteria pencarian yang diinginkan, kemudian tekan *enter* dan *cursor* akan berpindah ke nilai, diisikan dengan nilai yang diinginkan kemudian tekan tombol *enter*. Jika semua data yang diisikan berhasil maka akan tampil pada layar), fungsi tambah data (pada bagian pengguna dapat menambah data. Jika pengguna ingin menambah data *propose demand* dapat memilih nomor pada lingkaran berwarna biru tua dan biru muda. Kemudian tekan simpan.), dan fungsi *reset* : digunakan untuk mengosongkan semua *textbox* yang terdapat di layar.

8.4 Tambah Baru Formulir Permohonan Pelatihan Khusus (FPPK)

Personnel Development

Formulir Permohonan Pelatihan Khusus (FPPK)

Personal Development > Training & Event Management > Baru FPPK

Baru Ubah/Hapus Tambah ITR Tampil

Cari Berdasarkan TOPIK

Nilai

Cari Topik

Daftar TOPIK & SUBTOPIK berdasarkan :
Ike %%

SEQ	ID_TPK	TOPIK	ID_STPK	SUBTOPIK
04	2	SAP	4	ENTITY RELATION DIAGRAM
05	2	SAP	5	CUSTOMER RELATIONSHIP
06	3	MLTM	6	VIDEO
02	1	SR	3	WORK UTILITIES
08	3	MLTM	8	FLASH
09	3	MLTM	9	DIMENSI
07	3	MLTM	7	AUDIO

FORMULIR ANALISIS KEBUTUHAN PELATIHAN

NO

ID_PROPOSE DEMAND

ID_TOPIK

TOPIK

ID_SUBTOPIK

SUBTOPIK

UNIT / PERUSAHAAN

ID_SEKSI

SEKSI / DEPARTEMEN

RENCANA PELAKSANAAN KEGIATAN PELATIHAN

FISCAL YEAR

START PERIODE

END PERIODE

TANGGAL

Tampilan di atas ini digunakan untuk tambah data pengajuan pelatihan khusus. Untuk menampilkan ini, pengguna harus login terlebih dahulu, kemudian pilih *company* dan *area* yang sesuai dengan hak aksesnya, kemudian pilih menu *personnel development* → *propose demand* → *fppk*. Fungsi – fungsi yang disediakan untuk memudahkan pengguna adalah fungsi pencarian (pengguna cukup memilih kriteria pencarian yang diinginkan, kemudian tekan *enter* dan *cursor* akan berpindah ke nilai, diisikan dengan nilai yang diinginkan kemudian tekan tombol

enter. Jika semua data yang diisikan berhasil maka akan tampil pada layar), fungsi tambah data(pada bagian pengguna dapat menambah data. Jika pengguna ingin menambah data *fppk* dapat memilih nomor pada lingkaran berwarna biru tua dan biru muda. Kemudian tekan simpan.), dan fungsi *reset* (digunakan untuk mengosongkan semua *textbox* yang terdapat di layar).

8.5 Tampil Business Event Plan

Business Event Plan (BEP)

Personal Development > Training & Event Management > Ubah & Hapus BEP

Cari Berdasarkan: SUBTOPIK
Nilai:
Cari DATA

Daftar Propose Demand berdasarkan :
Ike %%

UNcheck check	NO	TOPIK - SUBTOPIK PELATIHAN	TANGGAL PELAKSANAAN	JML PESERTA BERDASARKAN JABATAN**			INSTRUKTUR		
				OP	OL	SV	NIK	NAMA	JABATAN (DEPT/SEKSI)
<input type="checkbox"/>	1	SAP - ENTITY RELATION DIAGRAM	05/06/2008	15	16	17	99900	VALENTINO AM	NP -
<input type="checkbox"/>	2	SAP - ENTITY RELATION DIAGRAM	05/06/2008	15	16	17	99991	YUNI	NP -
<input type="checkbox"/>	3	SAP - ENTITY RELATION DIAGRAM	05/06/2008	15	16	17	99992	WULANSARI	NP -
<input type="checkbox"/>	4	SAP - ENTITY RELATION DIAGRAM	05/06/2008	15	16	17	10999	MARESHA CAROLINE	NP -
<input type="checkbox"/>	5	SAP - ENTITY RELATION DIAGRAM	05/06/2008	15	16	17	99991	YUNI	NP -
<input type="checkbox"/>	6	SAP - ENTITY RELATION DIAGRAM	05/06/2008	15	16	17	99992	WULANSARI	NP -
<input type="checkbox"/>	7	SAP - ENTITY RELATION DIAGRAM	05/06/2008	15	16	17	10002	MARCELLINE FENDRIANTO	NP -

Tampilan ini digunakan untuk melihat *bep res* yang terdapat dalam table di *database*.

8.6 Tampil Attendee

Attendee Business Event Plan (RES ATT)

Personal Development > Training & Event Management > Tampil ATT & BE

Cari Berdasarkan: TOPIK BEP
Nilai:
Cari DATA

Daftar BEP & RES berdasarkan :
Ike %%

Print Preview Download

UNcheck check	ID PREAT	ID F BEP	TPK	STP	TAAW	TAAK	EMP#	NAME	COMPANY	AREA	PHONE	KETERANGAN	STATUS
<input checked="" type="checkbox"/>	1	2	SAP	CUSTOMER RELATIONSHIP	23/06/2008 12:00	23/06/2008 10:00	99991	YUNI	PIC1	PIC1	-	-	Jadi
<input checked="" type="checkbox"/>	2	2	SAP	CUSTOMER RELATIONSHIP	23/06/2008 10:00	23/06/2008 09:00	10002	MARCELLINE FENDRIANTO	PF12	PF12	02270993322	-	Jadi
<input checked="" type="checkbox"/>	3	2	SAP	CUSTOMER RELATIONSHIP	23/06/2008 10:00	23/06/2008 09:00	99900	VALENTINO AM	PIT1	PIT2	02270462032	-	Jadi
<input checked="" type="checkbox"/>	4	2	SAP	CUSTOMER RELATIONSHIP	23/06/2008 10:00	23/06/2008 09:00	99992	WULANSARI	PIT1	PIT1	022081320210333	-	Jadi
<input checked="" type="checkbox"/>	5	7	MLTM VIDEO		03/06/2008 10:00	03/06/2008 09:00	10001	EVLIN MARCELLINE	PF12	PF11	08122003322	-	Tunggu
<input checked="" type="checkbox"/>	6	7	MLTM VIDEO		03/06/2008 10:00	03/06/2008 09:00	10002	MARCELLINE FENDRIANTO	PF12	PF12	02270993322	-	Batal
<input checked="" type="checkbox"/>	7	7	MLTM VIDEO		03/06/2008 10:00	03/06/2008 09:00	99992	WULANSARI	PIT1	PIT1	022081320210333	-	Batal

Print Preview Download

Tampilan ini digunakan untuk melihat *data attendee*. Fungsi – fungsi yang disediakan adalah pengurutan, pemilihan data, download data dan pencarian data.

8.7 Display Master Data Room

ID LOCATION	ID BUILDING	SHORT NAME	LONG NAME	TELEPHONE	FAX	CONTACT PERSON	EMAIL ADDRESS	WEBLINK	KETERANGAN
LOC_I_0001	BUI_I_0001	VC	VINSEN COMPUTER	08122002233	84467377	-	vin_2233@yahoo.com	http://www.vin-comp.com	-
LOC_I_0001	BUI_I_0002	TM	TOKO MARESHA	081931160037	91510705	-	shasha_0911@yahoo.com	http://www.sha.com	-
LOC_I_0002	BUI_I_0002	ST	SUGIANTO SHOP	0	0	-	nama@alamat.eks	http://	-
LOC_I_0002	BUI_I_0004	US	USEP SHOP	0	0	-	nama@alamat.eks	http://	-
LOC_I_0002	BUI_I_0005	TS	TATANG SHOP	0	0	-	nama@alamat.eks	http://	-
LOC_I_0002	BUI_I_0006	VS	VENTI SHOP	0	0	-	nama@alamat.eks	http://	-
LOC_I_0002	BUI_I_0007	OS	OPIK SHOP	0	0	-	nama@alamat.eks	http://	-

Tampilan di atas ini digunakan untuk menampilkan data *room* yang terdapat di dalam tabel. Fungsi – fungsi yang disediakan adalah pengurutan, pemilihan data, download data dan pencarian data.

9 Kesimpulan

Kesimpulan yang didapat dari keseluruhan web ini yaitu secara umum aplikasi ini menghasilkan nilai guna yang cukup tinggi, dimana aplikasi ini dapat memberikan solusi pada masalah yang terjadi pada sistem pelatihan sebelumnya (HRPuzzle) bahkan dapat meningkatkan efisiensi pekerjaan (*transformasi form manual- terkomputerisasi*). Beberapa hal yang ditawarkan dari aplikasi ini terhadap pengguna adalah kemudahan untuk melakukan pencarian dengan disediakannya kategori pencarian pada tiap halamannya, keamanan dalam mengakses data untuk pegawai *level 1 – level 8* karena ada otorisasi *role*, kemudahan untuk mengetahui apa saja yang terjadi di lapangan tanpa harus datang ke lapangan, kemudahan untuk mengakses data dengan adanya fitur *autocomplete* dan kemudahan untuk mengetahui *event* apa saja yang akan, sedang atau belum terjadi.

Fungsi enkripsi, *autocomplete*, dan *role* pada pembangunan web ini sangat memberikan nilai tambah bagi aplikasi yang telah berjalan ini sehingga tidak ada orang yang dapat mengetahui *password* satu dengan yang lainnya sekalipun *administrator*. Jika sampai terjadi lupa *password* dapat mereset dengan menghubungi *administrator*.

12 Saran-saran

Saran yang diberikan untuk mengembangkan aplikasi lebih lanjut yaitu: aplikasi ini akan lebih baik jika ditambah dengan pengaturan keuangan pelatihan terintegrasi dan fitur sms untuk pengajuan pelatihan. Pengoptimalan aplikasi web ini dapat dilakukan dengan menambahkan fitur keuangan pada aplikasi yang ada dan penambahan fitur pengajuan pelatihan pada *business event*.

Daftar Pustaka

- [1] A, Silberschatz, H.F Korth, S. Sudarshan, *Database Systems Concept*, McGraw Hill Companies, New York, 1997.

- [2] Alivia Yulfitri, "Proses Bisnis", available from: <http://pipiew.wordpress.com/2007/11/29/proses-bisnis>, diakses tanggal 10 Februari 2008.
- [3] Arif Mursodo, "SAP (System Application Product in data processing)", available from : <http://www.caaip.net>, diakses tanggal 10 Februari 2008.
- [4] Constantianus, Frederick, Bernard Renaldy Sutedja (2005). Analisa dan Desain Sistem Bimbingan Tugas Akhir Berbasis Web dengan Studi Kasus Fakultas Teknologi Informasi. Jurnal Informatika Universitas Kristen Maranatha Vol. I, No. 2, Desember 2005 : 93 - 106.
- [5] Dennis, Alan. Barbara Halley Wixom, Roberta M. Roth. Systems Analysis Design Third Edition. Von Hoffmann, Inc. 2006
- [6] "Data Flow Diagram (DFD)", available from: <http://library.Gunadarma.ac.id/files/disk1/2/jbptgunadarma-gdl-course-2004-imamahmadt-66-perancis-a.pdf>, diakses tanggal 12 Februari 2008.
- [7] "Entity-relationship diagram" available from: <http://www.techtarget.com/>, diakses tanggal 13 Februari 2008.
- [8] Greant Zak, Graeme Merrill, Torben Wilson, Brett Michlitsch. PHP Functions Essential Reference. Penerbit New Riders Publishing. Indiana. 2002.
- [9] Imbar, Radiant Victor. Bernard R Suteja. Pemrograman Web Commerce dengan *Oracle* dan ASP. Penerbit Informatika. Bandung. 2006
- [10] Imbar, Radiant Victor, *Materi DFD*. Bandung. Universitas Kristen Maranatha.
- [11] Imbar, Radiant Victor, *Materi Perkuliahan Basis Data Praktikum*. Bandung. Universitas Kristen Maranatha.
- [12] Prasetyo, Didik Dwi. 101 Tip & Trik Pemrograman Php. 2006. PT Elex Media Komputindo PT Gramedia, Jakarta. ISBN 979-20-8367-7
- [13] Putra, Dewanto Adi, Radiant Victor Imbar (2007). Perangkat Lunak Pengelolaan Informasi Data Pelatihan dan Aplikasi untuk Rekomendasi Nama Peserta Pelatihan dengan Studi Kasus di BPP-BSDM. Jurnal Informatika Universitas Kristen Maranatha Vol. II, No. 2 September 2007 : 167 - 182.
- [14] R.S. Pressman & Associates, Inc. Jakarta, 2008
- [15] Sutedja, Bernard Renaldy, Dkk, *Mudah dan Cepat Menguasai Pemrograman Web*, Penerbit Andi Offset, Yogyakarta, 1995.
- [16] Steven Feurstein. Bill Pribyl. Debby Russell: *Oracle PL / SQL Programming*. 1997.
- [17] Tirta, Eric, Radiant Victor Imbar (2007). Analisa, Perancangan dan Implementasi Sistem Informasi Penjualan Pelumas Studi Kasus : Perusahaan "PT. Pro Roll International". Jurnal Informatika Universitas Kristen Maranatha Vol. III, No. 1 Juni 2007 : 119 - 149.
- [18] Yeliana, Elisabet Setiawan (2007). Aplikasi Mobile Pembelian Handphone, Aksesoris Handphone dan Voucher Elektronik dengan Penggunaan GPRS dengan Studi Kasus Pada Toko Handphone dan Aksesoris X'SIST COMMUNICATION. Jurnal Informatika Universitas Kristen Maranatha Vol. II, No. 2 September 2007 : 137 – 152.

PEDOMAN PENULISAN ARTIKEL

Jurnal Sistem Informasi UKM menerima karya tulis:

- Dalam bentuk hasil penelitian , tinjauan pustaka, dan laporan kasus dalam bidang ilmu yang berhubungan dengan Teknologi Informasi khususnya dibidang Sistem Informasi.
- Belum pernah dipublikasikan dalam jurnal ilmiah manapun. Bila pernah dipresentasikan, sertakan keterangan acara, tempat, dan tanggalnya.
- Ditulis dalam bahasa Indonesia atau bahasa Inggris.

Sistematika yang ditetapkan untuk tiap kategori karya-karya tulis tersebut adalah:

1. Artikel Penelitian :

Hasil penelitian terdiri atas judul, penulis, abstrak berbahasa Indonesia untuk artikel berbahasa Inggris atau abstrak berbahasa Inggris untuk artikel berbahasa Indonesia (masing-masing terdiri atas 150-200 kata), disertai kata kuncinya. Pendahuluan, metoda, pembahasan, simpulan, dan saran, serta daftar pustaka (merujuk sekurang-kurangnya 3 [tiga] pustaka terbaru).

2. Tinjauan Pustaka:

Naskah hasil studi literatur terdiri atas judul dan penulis. Pendahuluan (disertai pokok-pokok ide kemajuan pengetahuan terakhir sehubungan dengan masalah yang digali). Permasalahan mencakup rangkuman sistematik dari berbagai narasumber. Pembahasan memuat ulasan dan sintesis ide. Simpulan dan saran disajikan sebelum daftar pustaka. Tinjauan pustaka merujuk pada sekurang-kurangnya 3 (tiga) sumber pustaka terbaru.

3. Laporan Kasus:

Naskah laporan kasus terdiri atas judul, abstrak berbahasa Indonesia untuk teks artikel berbahasa Inggris atau abstrak berbahasa Inggris untuk teks artikel berbahasa Indonesia (50-100 kata) disertai kata kuncinya, pendahuluan (disertai karakteristik lokasi, gambaran umum budaya yang relevan, dll), masalah, pembahasan, dan resume atau simpulan.

Tatacara penulisan naskah:

- a. Artikel diketik rapi dengan menggunakan Microsoft Word, dikirim dalam disket beserta print-outnya. Jenis huruf yang digunakan adalah *Cambria/Times News Roman* ukuran 11. Panjang artikel berkisar 10 – 11 halaman, ukuran kertas B5, satu spasi. Judul ditulis di tengah-tengah ukuran 14.
- b. Artikel ditulis dalam bahasa Indonesia atau bahasa Inggris yang baik dan benar. Abstrak ditulis miring (*italic*) ukuran huruf 11. Panjang gambar dan foto harus dalam bentuk jadi dengan resolusi gambar yang memadai (jelas dan

nyaman dilihat), serta dalam ukuran yang sesuai dengan format jurnal ilmiah, dan dalam bentuk disket.

- c. Daftar pustaka ditulis alfabetis sesuai dengan nama akhir (tanpa gelar akademik) baik penulis asing maupun penulis Indonesia, berisi maksimal 15 (lima belas) penulis yang dirujuk, font ukuran 11.
- d. Penulis mencantumkan institusi asal dan alamat korespondensi lengkap. Penulis yang artikelnya dimuat akan mendapat imbalan/honor peserta beserta 2 eksemplar jurnal ilmiah.
- e. Kepastian pemuatan atau penolakan akan diberitahukan secara tertulis. Artikel yang tidak dimuat akan dikembalikan. Redaksi jurnal ilmiah berhak melakukan penyuntingan.

Tatacara penulisan referensi/daftar pustaka :

Mengacu pada format American Psychological Association (APA)

1. Buku

a. Buku tanpa Bab

Referensi pada tulisan

. . . which offered a theoretical backdrop for a number of innovative behavior modification approaches (Skinner, 1969).

Referensi pada akhir tulisan (daftar pustaka)

Skinner, B.F. (1969). *Contingencies of reinforcement*. New York: Appleton-Century-Crofts. Bremner, G., & Fogel, A. (Eds.). (2001). *Blackwell handbook of infant development*. Malden, MA: Blackwell.

b. Buku dengan Bab

Referensi pada tulisan

. . . The elucidation of the potency of infant-mother relationships, showing how later adaptations echo the quality of early interpersonal experiences (Harlow, 1958, chap. 8).

Referensi pada akhir tulisan (daftar pustaka) Harlow, H. F. (1958). Biological and biochemical basis of behavior. In D. C. Spencer (Ed.), *Symposium on interdisciplinary research* (pp. 239-252). Madison: University of Wisconsin Press.

c. Buku tanpa penulis

Referensi pada tulisan

. . . the number of recent graduates from art schools in France has shown that this is a trend worldwide (Art Students International, 1988).

Referensi pada akhir tulisan (daftar pustaka)

Art students international. (1988). Princeton, NJ: Educational Publications International.

d. Buku dengan edisi / versi

Strunk, W., Jr., & White, E. B. (1979). *The elements of style* (3rd ed.). New York: Macmillan.

Cohen, J. (1977). *Manual labor and dream analysis* (Rev. ed.). New York: Paradise Press.

American Psychiatric Association. (1994). *Diagnostic and statistical manual of mental disorders* (4th Ed.). Washington, DC: Author.

e. Buku terjemahan

Luria, A. R. (1969). *The mind of a mnemonist* (L. Solotaroff, Trans.). New York: Avon Books. (Original work published 1965)

f. Buku dengan beberapa volume

Referensi pada tulisan . . . The cognitive development of the characters in Karlin's class illustrates the validity of this new method of testing (Wilson & Fraser, 1988-1990).

Referensi pada akhir tulisan (daftar pustaka) Wilson, J. G., & Fraser, F. (Eds.). (1988-1990). *Handbook of wizards* (Vols. 1-4). New York: Plenum Press.

2. Jurnal

a. Artikel Jurnal

Referensi pada tulisan When quoting an author's words exactly, indicate the page number: Even some psychologists have expressed the fear that "psychology is in danger of losing its status as an independent body of knowledge" (Peele, 1981, p. 807).

Referensi pada akhir tulisan (daftar pustaka) Peele, S. (1981). Reductionism in the psychology of the eighties: Can biochemistry eliminate addiction, mental illness, and pain? *American Psychologist*, 36, 807-818.

b. Artikel Jurnal, lebih dari enam pengarang

Referensi pada tulisan

. . . the nutritional value of figs is greatly enhanced by combining them with the others (Cates et al., 1991).

Referensi pada akhir tulisan (daftar pustaka)

Cates, A. R., Harris, D. L., Boswell, W., Jameson, W. L., Yee, C., Peters, A. V., et al. (1991). Figs and dates and their benefits. *Food Studies Quarterly*, 11, 482-489.

3. Sumber Digital

a. Buku elektronik dari perpustakaan digital

Wharton, E. (1996). *The age of innocence*. Charlottesville, VA: University of Virginia Library. Retrieved March 6, 2001, from netLibrary database.

b. Artikel Jurnal dari perpustakaan digital

Schraw, G., & Graham, T. (1997). Helping gifted students develop metacognitive awareness. *Roeper Review*, 20, 4-8. Retrieved November 4, 1998, from Expanded Academic ASAP database.

c. Artikel Majalah atau Koran dari Internet (bukan dari perpustakaan digital)

Sarewitz, D., & Pielke, R. (2000, July). Breaking the global warming gridlock [Electronic version]. *The Atlantic Monthly*, 286(1), 54-64.

d. Artikel e-Journal

Bilton, P. (2000, January). Another island, another story: A source for Shakespeare's *The Tempest*. *Renaissance Forum*, 5(1). Retrieved August 28, 2001, from <http://www.hull.ac.uk/renforum/current.htm>

e. Halaman Web

Shackelford, W. (2000). The six stages of cultural competence. In *Diversity central: Learning*. Retrieved April 16, 2000, from http://www.diversityhotwire.com/learning/cultural_insights.html

f. Web Site dari organisasi

American Psychological Association. (n.d.) *APAStyle.org: Electronic references*. Retrieved August 31, 2001, from <http://www.apa.org/journals/webref.html>

4. Sumber Lain

a. Artikel Koran, tanpa pengarang

Counseling foreign students. (1982, April). *Boston Globe*, p. B14.

b. Tesis

Caravaggio, Q. T. (1992). *Trance and clay therapy*. Unpublished master's thesis, Lesley University, Cambridge, MA.

c. Desertasi

Arbor, C.F. (1995). *Early intervention strategies for adolescents*. Unpublished doctoral dissertation, University of Massachusetts at Amherst.

Keterangan lain yang diperlukan dapat diperoleh dengan menghubungi redaksi melalui:

Sekretariat Jurnal Sistem Informasi UKM
Jurusan Sistem Informasi, Fakultas Teknologi Informasi
Universitas Kristen Maranatha
Jl. Prof. Drg. Suria Sumantri, MPH, No. 65 Bandung. 40164
Telp (022) 2012186, Fax (022) 2015154
Email: jurnal.si@itmaranatha.org
Website: <http://www.itmaranatha.org/jurnal/jurnal.sistem-informasi>

FORMULIR BERLANGGANAN

1. Nama :

2. Alamat :

3. Telepon/HP :

4. Email :

Menyatakan untuk berlangganan Jurnal Informatika mulai Edisi :
..... dan bersedia membayar biaya cetak dan ongkos kirim
sebesar Rp. 50.000 (/eks).

Biaya akan dikirim ke rek. **613-130-10005-2** ,NISP Bandung a/n **Radiant
Victor Imbar/Elisabet**

Pemohon :

(.....)

- Formulir Berlangganan dan Bukti Transfer dapat dikirim lewat pos/faks/email ke :
 - Universitas Kristen Maranatha
 - Fakultas Teknologi Informasi (FIT)
 - Alamat : Jl. Suria Sumantri 65 Bandung – 40164
 - Faks : +62-022- 2005915
 - Email : jurnal.si@itmaranatha.org